Smart proximity based service agents on mobile handsets

Lim, Luo Ren

2013

Lim, L. R. (2013). Smart proximity based service agents on mobile handsets. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54867>

<https://doi.org/10.32657/10356/54867>

SMART PROXIMITY BASED SERVICE AGENTS ON MOBILE HANDSETS

LIM LUO REN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2013

Acknowledgments

The work presented in this thesis was carried out under the supervision of Assoc. Prof Gan Woon Seng, Head of Division of Nanyang Technological University, School of Electrical and Electronic Engineering (EEE), Division of Information Engineering, Singapore and former supervisor, Dr. Quah Tong Seng. I would like to express my sincere gratitude for their guidance and motivation that had been instrumental in the completion of this research work.

I would also like to express my appreciation to Mr. Foon Kwok Heng, Johnson from the InfoComm Research Laboratory and Mr Lim Boon Song from the ICIS Practicum Laboratory for their technical and administrative support. The research carried out in this thesis was supported by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research grant NRF2007IDM-IDM002-080.

Last but not least, I thank my family and friends especially to my beloved fiancée, Ai Jia for their unwavering support and understanding throughout the course of my study.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	I
SUMMARY.....	IV
LIST OF FIGURES.....	V
LIST OF TABLES	VII
LIST OF ACRONYMS.....	VIII

CHAPTER 1

INTRODUCTION	1
1.1 MOTIVATION	3
1.2 OBJECTIVES.....	4
1.3 MAJOR CONTRIBUTION OF THE THESIS.....	5
1.4 ORGANIZATION OF THE THESIS.....	7

CHAPTER 2

LOCALIZATION TECHNIQUES AND RELATED WORK	9
2.1 INTRODUCTION.....	9
2.2 SIGNAL PROPERTIES	10
2.3 USE OF SIGNAL PROPERTIES IN LOCATION SYSTEMS.....	11
2.4 METHODS OF LOCALIZATIONS	12
2.4.1 DEAD RECKONING	14
2.4.2 LATERATION	14
2.4.3 ANGULATION	19
2.4.4 FINGERPRINT	21
2.5 MOBILE LOCATION SERVICES.....	24
2.5.1 MOBILE CELL IDENTIFICATION.....	25
2.5.2 SATELLITES.....	27
2.5.3 NETWORK	29
2.5.4 SHORT RANGE RADIO	31
2.6 IMPROVING LOCATION SENSING BY INTEGRATING POSITIONING TECHNOLOGIES	33
2.7 CONCLUSION	36

CHAPTER 3

ADAPTIVE CONTEXT MODEL FOR DISTRIBUTED EVENT BASED SYSTEMS.....	37
3.1 DIMENSIONS OF CONTEXT INFORMATION	40
3.1.1 TEMPORAL PROPERTIES	40
3.1.2 MULTIPLE REPRESENTATIONS	41
3.1.3 INTERRELATED DEPENDENCIES	41
3.1.4 IMPERFECT INFORMATION	41
3.2 CONTEXT SERVICE DELIVERY.....	42

3.3	CONTEXT REFINEMENT	44
3.3.1	QOC PARAMETERS	44
3.3.2	SUB PROCESSES	45
2.3.3	CONCEPTS AND IMPLEMENTATION	48
3.4	CONCLUSION	50
 CHAPTER 4		
WIRELESS LOCALIZATION FOR DISTRIBUTED MOBILE ENVIRONMENT		51
4.1	CONSIDERATION FACTORS IN MOBILE ENVIRONMENT	51
4.1.1	NOISE	51
4.1.2	CHANNEL INTERFERENCE	52
4.1.3	MULTIPATH EFFECTS	54
4.1.4	SIGNAL DRAG	55
4.2	MITIGATION STRATEGIES	64
4.2.1	PREVIOUS APPROACHES	64
4.2.2	ADAPTIVE CLUSTER SELECTION	65
4.3	CLUSTER BASED VS FILTER BASED	74
4.4	CONCLUSION	77
 CHAPTER 5		
PROXIMITY DISTRIBUTION FOR MOBILE APPLICATIONS		78
5.1	iSARIS DISTRIBUTION PLATFORM	79
5.2	IMPLEMENTATION ON WINDOWS MOBILE DEVICES	80
5.2.1	COMPONENT SYSTEMS	81
5.2.2	POSITION & DATA SERVER DESIGN	82
5.3	IMPLEMENTATION ON ANDROID DEVICES	83
5.3.1	BASIC ANDROID OPERATIONS	85
5.3.2	FUNCTION OVERVIEW	85
5.3.3	ACTIVITY CLASS	86
5.3.4	SERVICE AND SYSTEM CLASSES	88
5.3.5	SMART SPACE FUNDAMENTALS	89
5.3.6	DATABASE CLASSES	92
5.3.7	OTHER CLASSES	94
5.4	iSARIS SMART SPACE EFFICIENCY	96
5.5	CONCLUSION	98
 CONCLUSION AND FUTURE WORKS		99
AUTHOR'S PUBLICATIONS		101
REFERENCES		102

Summary

The availability of high performance smartphones had generated much interest in services that integrates computing capabilities into daily life. Research in this area focuses on making consumer devices context aware and appropriate to the circumstances. This thesis describes the development of an application distribution manager for non-persistent mobile application. We looked at different aspects of localization technologies available today and how these techniques are implemented in positioning systems. There is a gap in research on mobile service application distribution using location as the context. The thesis considers context modelling as a tool for improving ubiquitous computing. Context modelling is an important aspect in localization as it dictates the flow of information input into a system and the corresponding information outflow. Such a modelling tool can help develop a structured approach to handle location information. We describe a context refinement scheme that helps to manage context information while supporting user autonomy and control. We list the practical consideration on how the context refinement scheme can be applied to mobile localization based on the environment in which it operates. We looked at challenges associated with mobile environment and the proposals to mitigate such problems. Signal drag is a key issue identified to be resolved for the implementation of a proximity based mobile distribution service. A marked improvement in reducing signal drag in a wireless medium is observed using a proposed cluster based approach. The proposed cluster based solution is more efficient in identifying the relative contributions from wireless data. It is incorporated into the localization aspect of the proposed proximity distribution mechanism. The system is capable of distributing relevant service applications when in proximity to a smart space. The prototype proximity based software distribution platform tags applications to smart service areas.. Smart spaces are predefined areas that provide related service applications in the vicinity. This application distribution framework achieved context awareness by pushing only relevant applications to users when they are in a service area. The software prototype was tested in a real world setting using windows based and android based platforms and the results showed good recognition rate for areas of interest based on user's context and interest

List of Figures

FIGURE 1.1 USAGE PATTERN FOR DIFFERENT APPLICATION CLASSES. SHADED REGIONS REPRESENT APPLICATIONS THAT ARE NON-PERSISTENT. NON-PERSISTENT APPLICATIONS ARE EITHER SELDOM USED OR UNLIKELY TO BE RETAINED FOR THE LONG TERM OR A COMBINATION OF BOTH REASONS.	3
FIGURE 2.1 FUNCTIONAL BLOCK DIAGRAM OF A TYPICAL WIRELESS GEOLOCATION SYSTEM	13
FIGURE 2.2 TRILATERATION USING TOF/TOA.....	15
FIGURE 2.3 MULTILATERATION USING TDOA	17
FIGURE 2.4 TRIANGULATION USING AOA	20
FIGURE 2.5 RTT MEASUREMENT	27
FIGURE 2.6 GPS MODULATION SCHEME	28
FIGURE 2.7 EOTD SCHEME	31
FIGURE 3.1 CONTEXT DELIVERY FRAMEWORK.....	42
FIGURE 3.2 MODELLING MULTI-TIER CONTEXT OBJECTS (COS)	45
FIGURE 3.3 COMBINATION/EXTRAPOLATION	46
FIGURE 3.4 DEDUCTION	47
FIGURE 3.5 SELECTION.....	47
FIGURE 3.6 INTERPOLATION	48
FIGURE 3.7 FILTRATION	48
FIGURE 4.1 SIGNAL STRENGTH DISTRIBUTION PROFILE OF WLAN ACCESS POINTS AT A STATIC LOCATION. THIS INFORMATION IS USED TO CREATE SIGNAL FINGERPRINT SIGNATURE IN RSS BASED LOCALIZATION TO MAP PHYSICAL LOCATIONS IN KNOWN SPACE.	57
FIGURE 4.2 EFFECT OF TERMINAL ORIENTATION ON SIGNAL RECEPTION. SIGNAL DISTRIBUTIONS FOR EACH OF THE FOUR DIFFERENT ORIENTATIONS AT A STATIC POINT ARE COLLECTED USING 300 SAMPLES AT TWO SECOND INTERVAL FOR A SINGLE ACCESS POINT.	57
FIGURE 4.3 SIGNAL READINGS (AP A) ALONG THE WARDRIVING PATH IN A 1/2 MINUTE LAP.....	60
FIGURE 4.4 SIGNAL READINGS (AP B) ALONG THE WARDRIVING PATH IN A 1/2 MINUTE LAP	60
FIGURE 4.5 SIGNAL READINGS (AP C) ALONG THE WARDRIVING PATH IN A 1/2 MINUTE LAP	60
FIGURE 4.6 SIGNAL READINGS (AP A) ALONG THE WARDRIVING PATH IN A 1 MINUTE LAP.....	61
FIGURE 4.7 SIGNAL READINGS (AP B) ALONG THE WARDRIVING PATH IN A 1 MINUTE LAP	61
FIGURE 4.8 SIGNAL READINGS (AP C) ALONG THE WARDRIVING PATH IN A 1 MINUTE LAP	61
FIGURE 4.9 SIGNAL READINGS (AP A) ALONG THE WARDRIVING PATH IN A 2 MINUTE LAP.....	62
FIGURE 4.10 SIGNAL READINGS (AP B) ALONG THE WARDRIVING PATH IN A 2 MINUTE LAP	62
FIGURE 4.11 SIGNAL READINGS (AP C) ALONG THE WARDRIVING PATH IN A 2 MINUTE LAP	62
FIGURE 4.12 ESTIMATED LOCATIONS OF ACCESS POINTS A, B AND C (TOP TO BOTTOM). THE CIRCLE INDICATES THE REGION WHERE MOST INFLUENCE IS EXERTED ALONG THE MOVEMENT PATH DUE TO THE AP'S PROXIMITY. THE SUM VECTOR INDICATION FROM A SINGLE AP WITH RESPECT TO THE ACTUAL GROUND MOTION IS SHOWN USING THE ARROWS. OVERALL NET SIGNAL DRAG EFFECTS DEPENDS ON THE COMBINED VECTORS FROM ALL VISIBLE APs.	63
FIGURE 4.13 MAPPING LOCATION ESTIMATES USING FINGERPRINT SCHEMES AND ADAPTIVE RADIO MAP SELECTION. LOCATION ESTIMATES ARE PROVIDED BY SENSORS AND TRAINED DATA SETS.....	66
FIGURE 4.14 THREE-LAYER BACKPROPAGATION NEURAL NETWORK IMPLEMENTATION FOR NNSS PATTERN RECOGNITION AND CLASSIFICATION USING k NN. FIRST LAYER HANDLES INPUT DISTRIBUTION. SECOND LAYER PROVIDES FEEDBACK CONTROL. LAST LAYER FORMS NEURONS FOR OUTPUT CLASSIFICATIONS.....	67
FIGURE 4.15 COMPARISON OF VARIOUS Q SIZES ($Q=2,3,4$ FROM LEFT TO RIGHT) USED IN THE RADIO MAP CLUSTER ON COMPUTATION PERFORMANCE. THE VERTICAL AXIS REPRESENTS THE COMPUTATION TIME IN MILLISECONDS (MS). HORIZONTAL AXIS REPRESENTS THE AMOUNT OF EXECUTED RUNS.....	75
FIGURE 4.16 AVERAGE ESTIMATES ERROR IN SIGNAL MATCHING LOCALIZATION PERFORMANCE FOR FILTER AND CLUSTER BASED METHODS.....	76

FIGURE 4.17 CUMULATIVE ERROR DISTRIBUTION FUNCTIONS FOR CLUSTER, WCL, RSS AND GT BASED METHODS. AVERAGE DISTANCE ERROR DECREASED STATISTICALLY WITH THE USE OF CLUSTER BASED SOLUTION.....	76
FIGURE 5.1 MOBILE ENTERS A SR FROM PUBLIC AREA AS IT NEARS COMPLEX (LEFT). MOBILE ACCESSING SA DEFINED UNDER WI-FI COVERAGE (RIGHT).....	79
FIGURE 5.2 TYPICAL DEPLOYMENT AND INTERACTION BETWEEN ENTITIES.....	81
FIGURE 5.3 SHOPPING MALL SERVICE USING THE iSARIS PLATFORM.....	81
FIGURE 5.4 ALLOCATION OF SA ON POSITION SERVER.....	83
FIGURE 5.5 ANDROID ARCHITECTURE.....	84
FIGURE 5.6 CLASS DIAGRAM ILLUSTRATING THE DEPENDENCIES BETWEEN SYSTEM CLASSES.....	86
FIGURE 5.7 FLOW DIAGRAM ILLUSTRATING THE STATE PATHS OF ACTIVITY CLASSES.....	88
FIGURE 5.8 LOCATION DATA ARE SENT TO GOOGLE MAP SERVERS. DATA COLLECTED USING CROWDING SOURCING TECHNIQUES ARE USED TO BUILD GEOGRAPHIC ZONES BASED ON COMMON UNIQUE CELL IDS.	91
FIGURE 5.9 A LIST OF SUITABLE APPLICATIONS FOR THE PARTICULAR SERVICE AREA IS PUSHED TO THE DEVICE ON ENTRY INTO SMART SPACE SERVICE AREAS. APPLICATIONS ARE INSTALLED ON AND REMOVED FROM MOBILE DEVICE AS IT TRANSITS BETWEEN SPACES.	92
FIGURE 5.10 LIST OF SUITABLE APPLICATIONS FOR THE PARTICULAR SERVICE AREA IS PUSHED TO THE DEVICE ON ENTRY INTO SMART SPACE SERVICE AREAS. APPLICATIONS ARE INSTALLED ON AND REMOVED FROM MOBILE DEVICE AS IT TRANSITS BETWEEN SPACES.	95
FIGURE 5.11 OVERVIEW OF SMART SPACE CREATION AT TAKASHIMAYA, ORCHARD. ACCURACY OF SMART SPACE REPRESENTATION IMPROVES WHEN MORE CIRCLES ARE USED AS THEY APPROXIMATE THE LOCATION SHAPE BETTER.	96
FIGURE 5.12 AVERAGE APPLICATION RECOGNITION RATE IN EACH SMART SPACE CREATED BY VARYING NUMBER OF CIRCLES FOR A FEW LOCATIONS IN SINGAPORE.....	96
FIGURE 5.13 DIFFERENT RADII SIZE AFFECTS LOCATION RECOGNITION. IN GENERAL, SMALLER CIRCLE USED IN DATABASE FOR SA IS MORE IDEAL IN MATCHING MOBILE LOCATION SENSOR READINGS.	97

List of Tables

TABLE 1.1 COMPARISON OF CURRENT AND PROPOSED SYSTEMS	5
TABLE 2.1 WIRELESS LOCALIZATION APPLICATIONS.....	10
TABLE 2.2 RADIO SPECTRUM	11
TABLE 2.3 BASIC LOCALIZATION MEASUREMENT TECHNIQUES	13
TABLE 2.4 MOBILE TELEPHONY STANDARDS.....	24
TABLE 2.5 POSITIONING TECHNOLOGIES FOR MOBILE	25
TABLE 3.1 RATING DESCRIPTION FOR QoC PARAMETER TAGS	45
TABLE 3.2 QoC TAGS FOR CONTEXT INFORMATION	45
TABLE 3.3 DEDUCTION PROCESS EFFECT ON CO RANKING.....	49
TABLE 4.1 EFFECT OF Q ON CLUSTER COVERAGE FOR SEARCH REGION	74
TABLE 4.2 AVERAGE TIME TAKEN TO COMPLETE CLUSTER COMPUTATION (MS)	75
TABLE 5.1 DISTRIBUTABLE ANDROID APPLICATION LIST BY LOCATION ZONE.....	93

List of Acronyms

3GPP:	3G Partnership Project
ACAS:	Adaptive & Context-Aware Services
ACI:	Adjacent Channel Interference
AGPS:	Assisted Global Positioning System
AOA:	Angle of Arrival
AP:	Access Point
API:	Application Programming Interface
AWGN:	Additive White Gaussian Noise
BP:	Backpropagation
BSSID:	Basic Service Set Identifier
BTS:	Base Transceiver Station
C/A code:	Coarse/Acquisition code
CAS :	Context Aware Service
CCI:	Co-Channel Interference
CDMA:	Code Division Multiple Access
CGI:	Cell Global Identity
CID:	Cell Identification
CO:	Context Object
CP:	Context Provider
CMS:	Context Management Service
CNR:	Carrier-to-Noise Ratio
COBRA:	Common Object Request Broker Architecture
CVB:	Cumulative Virtual Banking
E-CID:	Enhanced Cell ID
EM:	Electromagnetic

EOTD: Enhanced Observed Time Difference

FAF: Floor Attenuation Factor

FM: Frequency Modulation

GLONASS: Global Navigation Satellite System (Russia)

GNSS : Global Navigation Satellite System (Generic)

GPS: Global Positioning System

GSM: Global System for Mobile Communications

GT: Ground Truth

GTD: Geometric Time Difference

HMM: Hidden Markov Model

ICMP: Internet Control Message Protocol

IDPL: Idle Period Downlink

IMEI: International Mobile Equipment Identity

IRNSS: Indian Regional Navigational Satellite System

iSARIS: Intelligent Service Area and Region Identification System

k NN: k -Nearest Neighbors

LAC: Location Area Code

LBS: Location Based Service

LMU: Location Measurement Unit

LOS: Line-Of-Sight

MAC: Media Access Control

MAP: Maximum A Posterior

MCC: Mobile Country Code

MNC: Mobile Network Code

MSSSD: Mean Square Signal Strength Deviation

MUSE: Multi-use Sensor Environment

NLOS: Non-Line-Of-Sight

NMR: Network Management Records

NNSS: Nearest Neighbor in Signal Space

OFDM: Orthogonal Frequency-Division Multiplexing

OSI: Open Systems Interconnection

OTD: Observed Time Difference

OTDOA: Observed Time Difference of Arrival

P-code : Precision code

QoC: Quality of Context

QZSS: Quasi-Zenith Satellite System

RCPI : Received Channel Power Indicator

RF: Radio-Frequency

RFID: Radio-Frequency Identification

RNC: Radio Network Controller

RSPB: Reference Signal Power Budget

RSS: Received Signal Strength

RSSI: Received Signal Strength Indication

RTD: Real-Time Difference

RTT: Round Trip Time

RXLEV: Received Signal Level

SA: Service Area

SDE: Software Development Environment

SDEM: Signal Dragging Error Metric

SDK: Software Development Kit

SNR: Signal to Noise Ratio

SR: Service Region

SS: Signal Strength

TA: Timing Advance

TA-IDPL: Time-aligned Idle Period Downlink

TCP: Transmission Control Protocol

TDOA: Time Difference of Arrival

TOA: Time of Arrival

TOF: Time of Flight

TTF: Time-To-first-Fix

UDP: User Datagram Protocol

UML: Unified Modeling Language

UMTS: Universal Mobile Telecommunications System

U-TDOA: Uplink Time Difference of Arrival

VM: Virtual Machine

WCDMA: Wideband Code Division Multiple Access

WCL: Weighted Centroid Localization

WLAN: Wireless Local Area Network

WPAN: Wireless Personal Area Network

XML: Extensible Markup Language

CHAPTER 1

Introduction

Mobile telephony today is no longer constrained to mere voice communications. As cellular networks expand in capacity and speed, the mobile phone has gradually become an alternative to the personal computer. Its use has expanded to include a variety of data and computing services, which opens up many pervasive computing possibilities for the modern day user. At the consumer level, cellular networks and wireless hotspots are more prevalent today due to lower costs of deployment and higher demand for reliable information network. These technologies provide avenues for spontaneous automated connectivity of users without the need for wired connections. With the convergence of mobile phone technologies, cheaper, smaller and faster computing devices can host new classes of mobile applications and services that revolutionize how user interacts with their surroundings. The expanding availability of high performance smartphones has led to increasing interest in services that integrates computing capabilities into daily life. Making consumer devices context aware and appropriate to the circumstances is an important research theme within the ubiquitous computing community.

Ubiquitous computing paradigm was first envisioned by Mark Weiser [1] depicting a world with non-intrusive availability of computing power throughout the physical environment which is effectively invisible to user. To harness computing power in the physical environment, one has to know where you are. Knowing where you are and how to get to your destination is just one aspect of mobile computing out of many other features that users would be interested in. Another key factor is managing user's interactions with entities that are in the immediate proximity. Ubiquitous computing effectively transform physical space around a user into digital "smart space" for human-computer interactions. Mobile application developers realized that location context forms an integral part of the overall pervasive computing experience. It also plays an important role in understanding whether the need for a particular mobile application is persistent to

the user. A non persistent mobile application loses its purpose when the mobile is physically moving away from the entity it is associated to.

A major development in location based service (LBS) technologies [2] is the gradual focus shift from reactive to proactive services. In order to obtain a mobile client's position, LBS software programmers require prior knowledge of the location application programming interfaces (APIs) for the relevant software development environment (SDE). With the help of these APIs, each service provider creates their own independent LBS products without the need to address vendor specific proprietary extensions or handle transition logic for different location context providers. This approach simplifies LBS programming by providing a standard location data retrieval about user location. Services that are automated by predefined events require less attention from user. Applications execute cross referencing with other services to facilitate multi-targets interaction. Interrelating positions of several targets improves situation awareness by collaborating contexts of various users. Previous focus of LBS was mostly on content delivery, often through the use of internet browser. With the introduction of internet enabled mobile applications, the delivery mechanism has become more dynamic and richer in its interaction with users. The internet and its content are no longer only accessible via the browser but also by user centric mobile applications which are purposely designed to match the needs of the users.

Developing LBS applications for mobile devices requires careful consideration of design challenges associated with mobility. Mobile devices have scarce resources such as low power capacity, slow processor and limited memory [3]. Temporary loss of connectivity is a norm rather than an anomaly, which inevitably causes connection session to be relatively shorter. System must be able to react to frequent unannounced changes such as variability in network bandwidth and resource availability. Requirements for mobile applications are considerably different from that of fixed distributed applications. A good mobile application design requires good understanding of user situations so as to support user autonomy and control [4]. Internet activities conducted through mobile devices often include information seeking, communication, transaction and content object handling [5]. Most of them are designed by user-centered, contextual or social computing approaches [6]. Instead of answering questions like "Where am I?" and "How to get there?", LBS today also provides answers to: "What can I do from here?" and "What are the interesting objects around?". LBS is no longer just mapping and navigation but also about tracking and augmented reality. These developments are the future for mobile applications.

1.1 Motivation

An important development in the mobile industry is the increase in number and variety of mobile applications that are available for the mobile devices. Current mobile application distribution channels, such as Apple App Store, Blackberry App World, and Google Play uses pull based schemes, where users request for content through a data service provider. Application distribution channels typically provide the software development kits (SDK) for developers to create their own mobile applications. These software are then uploaded to an online storage server and made available to the public for download onto their mobile phones. This storage, usually accessible by an application interface, has always adopted a pull based model for software content distribution. Users must actively access a database in order to find content and often aided by search tools or a recommender based on user ratings. Applications can also be organized into categories or featured by the distribution channel itself to gain exposure. Users access the distribution channel regularly to receive application updates. Such mobile software distribution techniques are the current norm for the mobile software industry.

Frequency of use	High	<p>Intense use over a finite period of time.</p> <p>Application no longer as useful after completion of content.</p> <p>E.g. Books, Games</p>	<p>Frequent use over long period of time.</p> <p>Content is constantly being refreshed to keep updated, providing consumers with constant value over time.</p> <p>Eg. News, Social Networks</p>
	Low	<p>Typically used only a few times and then cease its purpose.</p> <p>Value decrease significantly after initial use and seldom revisited.</p> <p>Eg. Travel, Lifestyle</p>	<p>Remains relevant for long period of time, but get used only occasionally</p> <p>Sustainable value and therefore consistently revisited over time.</p> <p>Eg. Medical, Government</p>
		Low	High
		Retention time	

Figure 1.1 Usage pattern for different application classes. Shaded regions represent applications that are non-persistent. Non-persistent applications are either seldom used or unlikely to be retained for the long term or a combination of both reasons.

A side effect of explosive increment in mobile applications is the difficulty faced by consumers in identifying the relevant applications that is applicable to their context. Mobile applications can be categorized by consumer usage patterns based on frequency of use and the amount of time it remains on device. Figure 1.1 shows how different applications are typically perceived by mobile user based on the nature of service provided.

Most users typically download an application at the moment of need or based on reviews from other users. Applications providing value added and time sensitive content tends to be used repeatedly and retained on the mobile device for long period of time. News, social networks and finance applications fit these criteria and are usually permanently installed on most smartphones. On the other hand, applications with limited durability like travel and lifestyle applications tend to lose consumer attention rapidly. The value of the information contained in such applications diminished significantly after the initial use. They are unlikely to be needed again anytime soon although it is possible that such application might be needed again from time to time. Most users do not utilize all their downloaded applications on a regularly basis. A significant portion of purchased applications were never used again a day after it was downloaded and usage decreases further with length of time after download. In this research, application classes indicated by the shaded region in Figure 1.1 are treated as non-persistent mobile applications (i.e. mobile applications that are either rarely used or retained). Such applications form a significant proportion of total mobile applications downloaded by users. As more such applications are developed, many of the older applications become obsolete. Users often only require the application at the moment of need and do not need necessary need to keep the applications for long term. Limited shelf life due to non-persistent applications is a cause of concern for application developers [7]. The area of research work in this thesis will be dealing with mobile applications and the distribution of such services using a proximity based service agent.

1.2 Objectives

The objective of this thesis is to develop an application distribution manager for non-persistent mobile application capable of distributing relevant service applications when in proximity to a smart space. Smart spaces are predefined areas that provide related service applications in the vicinity. This application distribution framework achieved context

awareness by way of pushing only related applications to users when they are within a service area. The proposed distribution mechanism addresses deficiency in current software distribution for non-persistent applications by automating software installation and removal processes. The system suggests mobile applications to the user for the smart space they are in by tagging targeted applications to selected smart service areas. These applications will be automatically removed from mobile device when they are not needed outside a service area. Thus, fewer applications are required to remain in the mobile device. To the best of my knowledge, current literature has not reported any proximity based distribution mechanism for mobile applications.

1.3 Major contribution of the Thesis

This thesis describes an alternate mobile application distribution mechanism that utilizes physical space as the preferred medium for user-application interaction. The proposed system is built on the Android platform with modifications made to software at the application framework level. The service area identification system uses cell tower based identification and wireless radio signature to spot smart spaces. Cell identification is combined with short range radio positioning methods to achieve efficient indoor localization. Table 1.1 shows a comparison of current technology deficiency against the proposed solution.

Table 1.1 Comparison of current and proposed systems

<i>Current Technique</i>	<i>Solution/ Benefits</i>
Application uses location APIs to retrieve location information.	Mobile phone monitors the vicinity of the user and triggers alerts when it identifies a smart service area.
Location services like GPS, Wi-Fi etc are called each time an application is used.	Monitoring of mobile cell beacon to track vicinity of user. Mobile cell location technique as an elementary location service and can upgrade to better techniques like GPS, etc, if required.
Start up time per API call is required.	Monitoring of cell beacons through the use of cell identification techniques requires little computing effort since a mobile phone need to monitor them as well.
Applications need to be updated manually every time a new version is available.	Applications that are downloaded will be the latest version as uploaded by the developers. All users will be using the same updated version.

Mobile applications providing value added and time sensitive content tend to be used repeatedly and retained on the mobile device for longer period of time. This class of applications is indicated by the non shaded region in Figure 1.1. News and finance applications fit into criteria. Such applications are not the priority of our proposed solution as they are inclined to be kept permanently installed by users. Instead, the proposed solution is treated as a supplement to such applications.

Current software distribution model suffers from problems such as those seen in Table 1.1. The solution presented in this thesis addresses this deficiency by relooking at how software distribution can be better managed . The major contributions of this thesis are as follows:

- 1) Instead of treating all mobile applications as equals, the new solution automate software installation and removal for mobile applications that are seldom retained by user. A user has the option to choose to install some or all of the applications from a recommendation list that are preselected and deemed as useful for the environment he is in. This method is more advantageous for user as compared to the conventional distribution method where user searches through an app store for the software he required. The thesis looks at how mobile applications should be treated base on the nature of the software.
- 2) A perpetual problem with current application distribution model is that mobile device is overloaded with applications that were not used on a frequent basis. Such a system takes up unnecessary resources takes produce a drag on performance. This thesis looks at methods to free up data storage resources on the smartphone by providing only context-relevant applications on the go. Applications are removed whenever a user leaves a designated smart space because research shows that such applications usually lose its usefulness once the user is outside the smart space. From the user perspective, the proposed system by this thesis helps in the organization of contents on the smartphone as it does away with applications that lose its usefulness when away from a stated locality.
- 3) The model describes in this thesis provides a faster way to develop applications by reducing the amount of time and effort to create codes for simple LBS. Location sensing activities is totally handled by the background location sensing service

mechanism without the need to rewrite location handling procedure for every application. Developers can include simple LBS functionality to any service without the need to rewrite LBS codes. This enables shorter time-to-market for service providers as compared to current model of distribution.

- 4) Due to the constant evolution in software lifecycle and quick development time, it is observed that mobile applications are updated more rapidly than ever before. For current distribution model, users requests for updates on a pull basis and we have situations where users may experience service disruption as their version of the software is not the latest. This is especially so for applications that are seldom used. Current method of distribution does not effectively allow newer version of mobile applications to reach the user in a timely manner. The thesis provides a solution to ensure that applications to end user is always the latest and compatible with other users in the same environment.
- 5) Last but not least, the thesis provides a set of protocol for context refinement. This context refinement protocol defines how quality of contextual information is captured and how it can be used to distinguish the importance of such contextual information to the user. The proposed solution improves the contextual information delivery process by consolidating contexts systematically. Such a system had not been implemented in the current literature.

1.4 Organization of the Thesis

The thesis is organized as follows:

Chapter 2 reviews literature and looks at the current methodologies of mobile application distribution. This chapter reviews the basic localization techniques and explains signal properties used in conventional systems. This explanation helps to give a basic understanding on the development of localization services in wireless and cellular networks, and how it can be applied in mobile distributed environment. It will also give a comprehensive critique on the current localization techniques used in proximity systems for similar distributed applications with a view to implement the best practices for the proposed software distribution model.

Chapter 3 describes context delivery in ubiquitous systems and proposes a means to differentiate between similar contexts. An objective context comparison model for computing systems is discussed. It describes a context refinement scheme that utilized a predefined set of quality attributes to achieve better location context comparisons and apply enhancements to source context. This systematic process helps improve context awareness in pervasive computing systems by tracking context information flows.

Chapter 4 looks at challenges associated with mobile operating in a wireless environment and its corresponding effects on localization. It lays out a strategy to minimize undesirable effects using a novel adaptive cluster selection scheme, which significantly improve wireless localization operations.

Chapter 5 describes the proximity based distribution system for mobile software and its implementation on windows mobile and android platform. The system is evaluated for its performance in creating smart space based on mobile proximity to various locations in Singapore.

Chapter 6 concludes the thesis and outlines future works for further investigation.

CHAPTER 2

Localization Techniques and Related Work

As technology improved, mobile users are no longer satisfied with just voice communication capability. As smartphones become more popular, the interest for mobile data service increases. There are a variety of ways to provide useful mobile services with location information being the most important class of contexts used by service providers. Such context aware data services are becoming the norm for smartphones. Context provisioning involves a variety of processes that encompass acquisition, refinement and dissemination of context data. The circumstances pertaining to the understanding of implicit situational awareness is essential to the use of context information. Context derived from a user situation exhibits properties that are different from user input or computed data. Mobile localization is seen to be a key area of research interest to support such services. A couple of interesting design problems remains to be resolved, such as improving current technologies, integrating positioning technologies and interoperability.

2.1 Introduction

The adoption of location information as a key context to realize ubiquitous computing services is evident. Many popular mobile applications utilize some form of wireless localization. User location is determined so as to provide content that is relevant to the current user situation at the venue. These LBSs usually deal with three main questions: Where is the user? What is around the vicinity? How to reach the desired destination?

Location information is often expressed as a spatial position or a textual description. A standard spatial coordinate system displays location as a latitude, longitude and altitude combination. For example, the World Geodetic System (WGS 84) datum describes latitude as 0-90 degrees north/south of the equator, longitude as between 0-180 degrees east/west of the prime meridian (the meridian passing through the Greenwich Observatory located in Greenwich, England) and altitude in meters above sea level. Textual description is used to associate places with either an address or a landmark, and is more ambiguous

than spatial location information. Many existing applications exploit location information to either search for products, create tour guides, offer direction guidance, match neighbouring devices, send mobile advertising or social networking [8]. The main classes of applications using wireless localization are shown in Table 2.1.

Table 2.1 Wireless Localization Applications

<i>Applications</i>	<i>Uses</i>	<i>Examples</i>
Navigation	Handling routes between locations	Road directions, Journey planning, Tour guide
Positioning	Identifying a spatial fix	Emergency services
Tracking	Monitoring movement of objects	People locator, Assets tracking, Social networking, Mobile advertising
Mapping	Creating graphical representation	Terrain mapping

2.2 Signal Properties

Many forms of signals including sound waves and certain segments of the electromagnetic (EM) spectrum are used in positioning technology [10, 43, 44]. Certain signal propagation properties, especially those in the radio bands are ideal for mobile localization as radio wave is the predominant means of wireless communication. The radio spectrum occupies the below 300 GHz part of the EM spectrum and is regulated by governments and international organizations, like International Telecommunication Union, International Special Committee on Radio Interference, European Telecommunications Standards Institute and European Conference of Postal and Telecommunications Administrations. The spectrum is divided into many different sub segments to be used in different radio transmission technologies. Ranges of allocated frequencies are often referred to by their use. The UHF and SHF spectrum attract the most research interest in the mobile domain. Table 2.2 shows the radio bands allocation and their use in communication technologies.

Table 2.2 Radio Spectrum

Band name	Symbol	ITU band	Frequency/ Wavelength	Communication usage
Extremely Low Frequency	ELF	1	3–30 Hz 100,000 km – 10,000 km	Deep sea submarine communication
Super Low Frequency	SLF	2	30–300 Hz 10,000 km – 1000 km	Deep sea submarine communication
Ultra Low Frequency	ULF	3	300–3000 Hz 1000 km – 100 km	Communication within underground mines
Very Low Frequency	VLF	4	3–30 kHz 100 km – 10 km	Near surface submarine communication, Navigation beacons
Low Frequency	LF	5	30–300 kHz 10 km – 1 km	Navigation (LORAN), AM longwave broadcasts (148.5 - 283.5 kHz) , amateur radio, RFID
Medium Frequency	MF	6	300–3000 kHz 1 km – 100 m	AM mediumwave broadcasts (EUR 531 kHz - 1611 kHz, AME 535 kHz - 1705 kHz)
High Frequency	HF	7	3–30 MHz 100 m – 10 m	AM shortwave broadcasts (2.3 MHz – 26.1MHz), amateur radio, over-the-horizon communications, RFID
Very High Frequency	VHF	8	30–300 MHz 10 m – 1 m	FM radio broadcasts, television broadcasts, line-of-sight ground-to-air, air-to-air communications, land and maritime communications
Ultra High Frequency	UHF	9	300–3000 MHz 1 m – 100 mm	Television broadcasts, digital audio broadcasts, mobile phones, Wi-Fi, Bluetooth, GPS, Ultra-HighFID
Super High Frequency*	SHF	10	3–30 GHz 100 mm – 10 mm	Wireless LAN, RADAR, satellite uplinks/ downlinks, terrestrial high-speed data links
Extremely High Frequency*	EHF	11	30–300 GHz 10 mm – 1 mm	High-speed data links, high-frequency microwave radio relay

*The SHF and EHF bands are sometimes referred to as microwave spectrum.

2.3 Use of Signal Properties in Location Systems

Common signal properties used in position detection and tracking include timing, strength, pattern and directionality [9]. Such properties are easily measurable in radio wave transmission during wireless communication.

Timing: The time-of-flight of a signal measures the time taken for a signal to traverse between two points at known velocity in a medium. The receiver observing the difference between transmission and arrival time of the emitted signal may itself be moving or stationary. A key criterion for time-of-flight systems is to ensure that the clock between sender and receiver is synchronized. Measuring time-of-flight of signals in the radio spectrum requires clocks with very high resolution, typically in the sixth orders of magnitude. A round-trip delay corresponding to twice the distance covered can be used to remove the need for any synchronization.

Strength: Signal attenuation is perhaps the most obvious characteristics of radio signal propagation. In general, signal strength depreciates predictably as distance between receiver and transmitter increases. This property held true in open environments with clear

Line-Of-Sight (LOS) propagation paths. Thus, it is possible to construct mathematical models to determine signal strength expected at different distances to receiver. Theoretically, indoor signal path loss obeys the distance power law [10],

$$P_r(d) = P_r(d_0) - 10n \log \left(\frac{d}{d_0} \right) + X_\sigma, \quad (1)$$

where P_r : received power,
 d_0 : reference distance,
 d : separation between RF signal receiver and transmitter,
 n : path loss exponent,
 X_σ : Gaussian random variable with zero mean and standard deviation, σ .

However, operating scenarios are often complicated by obstructions and multipath propagations. There are many radio propagation models [11] to predict average received signal strength in various environments. Research usually focused on giving prediction on the average RSS at any stated distance from transmitter in large scale propagation models. In addition, some research also focus on the variability of signal strength in close spatial proximity to any location for small scale or fading models.

Pattern: Some systems, especially those that operate in enclosed environment, rely on signal structure characteristics. Instead of trying to overcome the multipath phenomenon, multipath patterns are combined with other characteristics in signal propagation to create a pattern unique to any identified location. Specific service area can be divided into multiple signal pattern location grids. A signal signature database for each location grid can be generated in a prior calibration process. The system matches the signature to any entry in a predetermined database [12] to determine receiver's position.

Directionality: Location estimate can be made by computing the angle of each reference point in reference to the mobile entity. This process determines location by measuring angles from known points at a fixed baseline. The use of directionality to detect signals requires complex high cost antenna array. Such antennas can only be placed at the reference sites that is used to determine the position of the mobile entity and may not scale effectively when substantial mobile entities is involved.

2.4 Methods of Localizations

Generic localization system adopts a system, as seen in Figure 2.1. A list of localization techniques used in identifying location of a tracked object are displayed in Table 2.3 to show the characteristics of each method and the different forms of measurements required.

Figure 2.1 Functional block diagram of a typical wireless geolocation system

Table 2.3 Basic Localization Measurement Techniques

	Description	Signal Properties	Type
Dead Reckoning	Update previous measurements	None	Deterministic
Lateration	Distance measurements	Timing, Strength	Deterministic & Probabilistic
Angulation	Angle measurements	Directionality	Deterministic
Fingerprint	Radio signature	Strength, Pattern	Probabilistic
Basic Methods	Description	Limitations	
Proximity	Detects unidirectional presence of objects.	<ul style="list-style-type: none"> Does not give an exact fix. Detection limited to close distances. 	
Time-of-Arrival/Time-of-Flight (TOA/TOF)	Measures absolute travelling time of radio signal from a transmitter to receiver in order to determine travel distance	<ul style="list-style-type: none"> Require synchronization for both sensors & reference nodes Suitable for long range applications 	
Time-Difference-of-Arrival (TDOA)	Measures differences in signal arrival times at multiple receivers by determining the peak of the cross correlation output. The difference estimate defines a hyperboloid between receivers on which the mobile may exist	<ul style="list-style-type: none"> Only reference nodes requires synchronization Slightly poorer long range performance compared to TOA/TOF 	
Angle-of-Arrival (AOA)	Determines direction of propagation for radio frequency wave that is incident on an antenna array by measuring the phase and wave count of the received signal transmission.	<ul style="list-style-type: none"> Expensive due to the need to install directional antennas Subjected to multipath effects 	
Received Signal Strength (RSS)	Measures the RSS present in a radio transmission.	<ul style="list-style-type: none"> High accuracy for short range applications Accuracy deteriorates with increasing linear range 	

2.4.1 Dead Reckoning

Dead reckoning is probably the earliest methodology of navigation used to estimate current position based on previous determined position, known speed and acceleration, direction of movement, and elapsed time or distance travelled. This method depends very little on existing infrastructure and more on keeping track of previous movements. New position is estimated using previously determined initial position and the subsequent changes made to this position. However, because such estimate are poor in accounting for deviation, the error tends to accumulate with time and distance travelled. Such a system requires very precise entity tracking records to minimize mistakes.

2.4.2 Lateration

Lateration refers to the use of distance measurements to compute the position of an object using other reference points. Many iterations of this concept can be found in position determination systems.

Proximity: The simplest unilateration method detects presence of an object. The quality of the distance measurement is restricted to near or far. Alternatively, it can only answer a binary yes or no to a defined distance threshold. It is often applied to real-time locating technologies to observe processes from a distance. Proximity detection can also involve passive observation of signals in the environment to determine whether an object is in the vicinity of a known location. Detection of certain signals indicates relative proximity of the tracked object. The most fundamental form of proximity sensing is the physical contact with the object. Such technologies include pressure and touch sensors as well as capacitive field detectors. Another implementation example of proximity would be to monitor a mobile device to see if it is in range of access points in a wireless network. Proximity location sensing technique is also used in automatic identification systems. Radio-frequency identification (RFID) technology uses an object tag that can be applied to or incorporated into an entity for the purpose of identification and tracking using radio signals. Tags can be decipher from a distance and beyond card reader line-of-sight. RFID tags contain an integrated circuit and an antenna for receiving and transmitting signal information.

Time of Arrival (TOA)/ Time of Flight (TOF): Multiple distance measurements can be combined for a location fix. If the specific distances of a tracked object from 3 different

reference locations are known, trilateration can be applied to obtain a position fix on the object. TOA/TOF represents the time taken for electromagnetic wave to travel a distance through a medium, which in the case of mobile positioning systems, refers to the absolute travel time of a signal transmission from transmitter to receiver. Distance between transmitter and receiver can be directly calculated by utilizing the relationship between known velocity of signal in the medium and the frequency of the carrier signal. Position is located with distance measurements of coincident signals. When a signal leaves the transmitter, it arrives at two spatially separated receiver sites with a slight difference in time. Similarly, signals can be received at different times from separate transmitter sites even though it is sent simultaneously. The distances are established by measuring TOA from three separate sites to establish a location fix. By tracing all the possibilities of an object's position, an imaginary sphere around the reference point can be formed. A position estimate is derived by solving a trilateration problem, which calculates the intersection of the sphere surfaces with known centers and radii.

Figure 2.2 Trilateration using TOF/TOA

The three-dimensional trilateration problem is mathematically solved by calculating the intersection of the spheres created from extending known distances of the receiver object O from the respective reference points. A plane $z = 0$ is chosen such that all three spheres are on the same plane and centred. The sphere centers denoted by A, B, and C have radii of r_1 , r_2 , and r_3 respectively. Equations for all three spheres are given as:

$$r_1^2 = x^2 + y^2 + z^2, \quad (2)$$

$$r_2^2 = (x - d)^2 + y^2 + z^2, \quad (3)$$

$$r_3^2 = (x - i)^2 + (y - j)^2 + z^2. \quad (4)$$

The solution to the computed location would be the point (x, y, z) which satisfies all the spherical equations. Solution for x is given by:

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}. \quad (5)$$

Under normal circumstances, two spheres will intersect at more than a single point (i.e. $d - r_1 < r_2 < d + r_1$). Substituting x produces a circle equation that corresponds to the solution for the intersection. This can be imagined as a circular region made by the first and second sphere intersection, which also intersects with the third sphere, as stated

$$y^2 + z^2 = r_1^2 - \frac{(r_1^2 - r_2^2 + d^2)^2}{4d^2}. \quad (6)$$

Substituting $y^2 + z^2 = r_1^2 - x^2$ and solving for y and z yields:

$$y = \frac{r_1^2 - r_3^2 + (x - i)^2 - x^2 + j^2}{2j} = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x, \quad (7)$$

$$z = \pm \sqrt{r_1^2 - x^2 - y^2}. \quad (8)$$

Note that z can be a positive or negative square root. Therefore, it is possible to have zero, one or two solutions. If the circle does not come within the boundary of the third sphere, z will equate to the square root of a negative number, ie. no real solution exists. When a solution indicates a circle touches the surface of the third sphere at only one point, z equates to zero. When the solution circle touches the surface of the sphere at two separate points, z will be equal to a set of real solutions.

Time Difference of Arrival (TDOA): TDOA is the difference in signal arrival times at multiple receivers. A snapshot of the signal from a coincident source is taken at each of the receivers within a synchronized time period. A cross correlation of the signals is examined and the peak output gives the time difference for the arrival signal. A time difference estimate value creates a hyperboloid for two receivers on which a identified mobile can exist. Distances can be determined by the frequency and wave count of the signal transmission. Thus, the hyperboloid refers to a set of points which are all at a fixed range difference.

When the locations of two receivers are given, the same TDOA could be measured by a set of possible emitter locations located on a hyperboloid. A locus of all possible emitter locations forms a one half of a two-sheeted hyperboloid. Another hyperboloid is defined when this process is repeated with another receiver together with any previously used receiver. The intersection of the two hyperboloids will give position location estimate of the source. Location estimation is based on the intersection of all the hyperboloids. If a fourth receiver is used, a third TDOA measurement becomes available to be used as a measuring tool. The intersection of the third hyperboloid with the curve that was previously calculated with the first three receivers defines a point in space. In general, N receivers provide $N-1$ hyperboloids. Assuming a perfect model, in cases where $N > 4$ receivers are used, the $N-1$ hyperboloids should intersect only at a single point. Emitter's location would therefore be fully determined in a three dimensional space in an ideal model.

Figure 2.3 Multilateration using TDOA

Figure 2.3 illustrates a multilateration process that uses TDOA to estimate location. Consider a transmitter at unknown location (x,y,z) and four receiver sites at known locations A, B, C and D. The travel time of the pulse originating from emitters to each of the receiver locations, T , will be given by distance divided by signal propagation rate, c :

$$T_{A,B,C,D} = \frac{1}{c} \sqrt{(x - x_{A,B,C,D})^2 + (y - y_{A,B,C,D})^2 + (z - z_{A,B,C,D})^2}. \quad (9)$$

Taking site A as the origin, then

$$T_A = \frac{1}{c} \sqrt{x^2 + y^2 + z^2}. \quad (10)$$

The time difference of arrival, τ , between signals arriving directly at the central site and those originating via the peripheral sites can be shown to be:

$$\tau_B = T_B - T_A = \frac{1}{c} \left(\sqrt{(x - x'_B)^2 + (y - y'_B)^2 + (z - z'_B)^2} - \sqrt{x^2 + y^2 + z^2} \right), \quad (11)$$

$$\tau_C = T_C - T_A = \frac{1}{c} \left(\sqrt{(x - x'_C)^2 + (y - y'_C)^2 + (z - z'_C)^2} - \sqrt{x^2 + y^2 + z^2} \right), \quad (12)$$

$$\tau_D = T_D - T_A = \frac{1}{c} \left(\sqrt{(x - x'_D)^2 + (y - y'_D)^2 + (z - z'_D)^2} - \sqrt{x^2 + y^2 + z^2} \right), \quad (13)$$

where $(x'_{B,C,D}, y'_{B,C,D}, z'_{B,C,D})$ is the relative location of receiver sites B , C and D with reference to the origin which is located at site A .

Each equation defines a separate hyperboloid. As a result, the solution for the unidentified target location (x, y, z) , with reference to the origin at site A is obtained. In general, multilateration exhibits better accuracy in locating a target than sparse approaches with planar problems. Any errors in time measurement of the signals will affect the accuracy performance of the computed intersection. In practice, the surfaces rarely intersect perfectly at a single point but rather a rough estimated region in which the intersection may lie. Accuracy of multilateration is a complex function of numerous correlated variables such as:

- Antenna or sensor geometry of transmitter/receiver;
- Accuracy of the timing in receiver system;
- Frequency synchronization between transmitter and receiver oscillators;
- Phase synchronization of received signal with transmitted signal;
- Location information accuracy at known location.

Enhancement to accuracy is possible by using more receivers. TDOA of multiple transmitted signals can be averaged out to improve the accuracy of results. These steps reduce the location problem to an optimization problem for processing in computing systems.

2.4.3 Angulation

Angulation uses the direction of transmission to determine the object's location. The angles between tracked object and spatially separated reference sites give a geometrical solution to a position fix. The precision of angle measurement greatly influence the final outcome of the solution especially in situation where the receivers and transmitters are placed far apart.

Angle of Arrival (AOA): AOA determines propagation direction of any radio frequency waveform that falls incident on antenna array by measuring phase as well as wave count of the received signal transmission. To calculate an AOA, consider a two-element array spaced apart at a distance of half wavelength of a receiving RF wave. The wave could strike the array anywhere between the boresight and broadside resulting in a certain phase difference between the two antenna elements. This will indicate a corresponding AOA.

Angulation uses AOAs with respect to a baseline to compute an estimated position of an object. In multi-angulation, the travel time of a pair of synchronous radio signals from two transmitters is derived by measuring and computing the difference of distance with the relation of transmission velocity as an angle versus the baseline of the transmitters. A more common triangulation approach uses three separate points to estimate a position. Triangulation determines object location by measuring the angles of signal received from pre-identified points at the ends of a fixed baseline. The object location is fixed as the intersecting point of a triangle with a single known side and two known angles from two reference points on the baseline. Coordinates and distance is found by calculating the lengths of the triangle using trigonometric identities.

Figure 2.4 Triangulation using AOA

Let M be the position of the reference point which lies in between two receivers A and B . The receivers form a known distance baseline with M being the midpoint. Position C indicates the object being tracked. Assuming flat or Euclidean geometry, the distance RC is calculated by applying sine law to determine the side lengths in a triangle:

$$\frac{\sin \alpha}{BC} = \frac{\sin \beta}{AC} + \frac{\sin \gamma}{AB}. \quad (14)$$

Since distance AB is already identified and as such the lengths of the unknown two sides are:

$$AC = \frac{AB \cdot \sin \beta}{\sin \gamma}, BC = \frac{AB \cdot \sin \alpha}{\sin \gamma}. \quad (15)$$

RC can be derived by either $\sin \alpha$ or $\sin \beta$ using:

$$RC = AC \cdot \sin \alpha = BC \cdot \sin \beta. \quad (16)$$

Substituting AC or BC :

$$RC = \frac{AB \cdot \sin \beta \cdot \sin \alpha}{\sin \gamma} = \frac{AB \cdot \sin \alpha \cdot \sin \alpha}{\sin \gamma}. \quad (17)$$

The sum of angles in a triangle is 180° (i.e. $\alpha + \beta + \gamma = 180^\circ$). Using the trigonometric identity $\sin(\theta) = \sin(180^\circ - \theta)$,

$$\sin(\gamma) = \sin(\alpha + \beta), \quad (18)$$

which implies

$$RC = \frac{AB \cdot \sin \alpha \cdot \sin \beta}{\sin(\alpha + \beta)}. \quad (19)$$

This formula can be shown to be equivalent to $\sin(\gamma) = \sin(\alpha + \beta)$ using trigonometric identity,

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta. \quad (20)$$

Given AC or BC , point C coordinates is derived from the sine and cosine of the bearing from the observation point to calculate its offsets on the reference axis. The distance MC from the midpoint of AB to the unknown point C is calculated by determining MR and with the Pythagoras Theorem:

$$MR = AM - RB = \left(\frac{AB}{2}\right) - (BC \cdot \cos \beta), \quad (21)$$

$$MC = \sqrt{MR^2 + RC^2}. \quad (22)$$

Note that results may become inaccurate when the distances used are significantly large as compared to the curvature of the Earth. In such cases, spherical trigonometry should be adopted instead. Advance cellular systems could combine information about a mobile cell and timing to synchronize a mobile phone with the transceiver. Small aperture direction finder used in mobile networks would require complicated antenna arrays. In theory, numerous antenna arrays can cooperate to calculate the angle where a signal originated. The location of a targeted cell phone can be found using triangulation process by determining the respective angles of arrival at several cell sites.

2.4.4 Fingerprint

Received Signal Strength Indication (RSSI): RSSI is commonly implemented in Bluetooth, GSM/UMTS and Wi-Fi networks to monitor wireless network condition. It measures the relative RSS in a received radio signal transmission. RSSI value is usually masked from the user. Users can observe RSSI value using a wireless network monitoring tool. While RSSI is a rather poor estimator of link quality for wireless networks, it has been used frequently in RF fingerprint recognition system. RSSI is often correlated with

Received Channel Power Indicator (RCPI). RCPI is a functional measurement of the received RF power in a selected channel over the preamble and received frame. It covers the entire received frame with defined absolute levels of accuracy and resolution. These measurements are typically represented in dBm. RSSI is acquired during preamble stage.

The use of short range radio technologies for indoor positioning has been actively discussed [13, 14]. Related signal data records such as timing information, Access Points Media Access Control (MAC) addresses, signal strength, Signal-to-Noise Ratio (SNR) and Basic Service Set Identifier (BSSID) can be applied to check reliability and stability of radio signals from the Wireless Local Area Network (WLAN) infrastructure. This information can also act as useful parameters for mobile localization purposes. Empirical studies [13] had shown that an accuracy level of 1-3m is achievable in ideal conditions. In practice, the accuracy is hugely dependent on environment in which the RSS based mobile localization mechanism operates and the density of service nodes. Factors such as the presence of walls and other surfaces can attenuate signal propagation paths and presents a distorted signal signature presented to the receiver.

There are a few categories of systems that utilized short range radio positioning theory. Some of such localization approaches include nearest neighbor in signal space (NNSS) [15], neural networks [16], empirical methods[15, 20] and bayesian probabilistic networks [17, 18, 19]. In NNSS [15], a localization system measures minimum Euclidean distance between offline and real-time entries. A weighted average of k nearest neighbors are used to determine a location estimate based on the nearest result. Other ways include Multi layer perception neural networks [16]. An implementation such as a feed forward neural network that uses 3-input, 8-hidden, 2-output layers can achieve optimal generalization with low likelihood of overtraining. However, the relation between signal and location is hard to model. Another type of implementation uses Bayesian network to declare dependency relationships between random variables within a statistical distribution. One popular way to do this is to use maximum a posteriori (MAP) estimate for a bayesian probability model. This is implemented by assigning a probability for each possible location given any type of observation:

$$p(l | o) = \frac{p(o | l)p(l)}{p(o)} = \frac{p(o | l)p(l)}{\sum_{l' \in L} p(o | l')p(l')}, \quad (23)$$

where $p(o | l)$: likelihood function of obtaining observation o at location l
 $p(l)$: prior probability of location l

$p(o)$: normalizing constant

Recursive estimation is used to guess the last known position when tracking motion in radio positioning. A Kalman Filter [21] can be used to establish a linear relationship between state vector and measurements. This is because the relation is often clouded by random noise variations observed over time. For nonlinear problem, an Extended Kalman Filter [21] can be used. This, however, is not suited for large disturbances. Another way is to use Hidden Markov Model (HMM) [21, 22]. States are represented with separate HMM for every subset of state variable:

$$\lambda = \langle N, M, \{\Pi_i\}, \{a_{ij}\}, \{b_i(k)\} \rangle, \quad (24)$$

where N: number of states ($S_1, S_2, \dots, S_N$)
M: number of possible observations
 Π_i : initial probabilities, $P(q_0 = S_i)$
 a_{ij} : probabilities of state transition, $P(q_{t+1} = S_j | q_t = S_i)$
 $b_i(k)$: probabilities of observation, $P(O_t = k | q_t = S_i)$
O: observation
q: notation for a path of states

In order to identify the most probable path given any set of observations, Viterbi algorithm is used and is as:

$$\delta_t(i) = \operatorname{argmax} P(Q | O_1 O_2 \dots O_T) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_t = i, O_1 O_2 \dots O_t | \lambda]. \quad (25)$$

By rule of induction,

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}). \quad (26)$$

The most probable path to S_j would therefore have a penultimate state S_{i^*} where

$$i^* = \operatorname{argmax}_i \delta_t(i) a_{ij} b_j(O_{t+1}). \quad (27)$$

To choose a best path, the algorithm determines the possibility of taking on a new set of states. This is done by combining metric of previous state with incremental metric of next transition. Transition possibility from an old state to a new state will determine the incremental metric associated with a particular event.

2.5 Mobile Location Services

Mobile telephony is the provision of on-the-move communication services by connecting mobile phones to a telecommunication network consisting of transceiver stations tied to public switched telephone network. Mobile communication has evolved beyond traditional voice only services and moving to data throughput services. Table 2.4 shows the improvement in data technologies with each new generation of mobile telephony networks.

Table 2.4 Mobile Telephony Standards

Generation	Standards	Features
1G	<i>AMPS, TACS</i>	<ul style="list-style-type: none"> • Basic analog telecommunications
2G	<i>GSM, CDMA</i>	<ul style="list-style-type: none"> • digitally encrypted phone conversations • significant spectrum efficiency • small data services (SMS text messages)
2G (Transitional) [eg. 2.5G]	<i>HSCSD, GPRS, CDMA2000, EDGE</i>	<ul style="list-style-type: none"> • bridging system between 2G and 3G • enhanced data transmission • packet switching included in circuit switched domain
3G	<i>UMTS, WCDMA, CDMA2000</i>	<ul style="list-style-type: none"> • simultaneous use of data and speech services • high speed data rates • greater security
3G (Transitional) [eg. 3.5G]	<i>HSDPA</i>	<ul style="list-style-type: none"> • based on an all-IP network infrastructure • extensions to 3G standard • use advanced technologies eg. MIMO
4G	<i>LTE Advanced, WiMAX</i>	<ul style="list-style-type: none"> • provide comprehensive and secure all-IP based solution • standardization still in progress

The predominant positioning method used by mobile today is Global System for Mobile Communications (GSM) network. The newer Universal Mobile Telecommunications System (UMTS) network also provides features for positioning service. To standardize how positioning services are used in GSM networks, the Telecommunications Industry Association adopted cell identity and timing advance (Cell id + TA), enhanced observed time difference (EOTD), uplink time of arrival (TOA) and assisted GPS (AGPS) as common positioning standards for network operator. The 3G Partnership Project (3GPP) also adopted cell coverage, observed time difference of arrival (OTDOA) and AGPS for UMTS networks. Mobile applications can autonomously determine location using hardware modules on the mobile handsets or through the help of an external network [23]. In general, mobile positioning technologies can be classified into mobile-based, network-based and hybrid solutions. Table 2.5 summarizes the current mobile location technologies, and highlights the strengths and properties of each system. The techniques and their implementations are also described.

Table 2.5 Positioning technologies for mobile

	CID	E-CID	AOA	TDOA	EOTD	GPS	AGPS	Wi-Fi/ Bluetooth
Network			*					
Mobile	*	*				*	*	*
Hybrid				*	*		*	*

Technology	Attributes	Network	Handset	Accuracy
CID	Pilot measurements	All	Both	100m-3km (Depends on cell size and density)
E-CID	Additional time advance information	GSM only	Both	Default 500m. (Depends on bandwidth)
GPS/AGPS	GPS receivers in handsets or network	All	Upgrade	5-30m
AOA	Network measurements	All	Both	100-200m
U-TDOA	Network measures timing difference	All	Both	<50m
EOTD	Mobile measures timing difference	GSM only	Upgrade	50-200m
Wi-Fi/ Bluetooth	Signal pattern recognition complement	All	Upgrade	1-3m (Depends on node density)

2.5.1 Mobile Cell Identification

This concept works on the fact that mobile phone is often connected to the closest cell towers [24, 25]. Although mobile phone networks are optimized for capacity and call handling rather than localization, it can still be a very good starting point for position determination processes. In this approach, the mobile is assumed to be at the service cell's antenna position or the centre of a sector cell.

Cell Global Identity (CGI) is a unique global identity to address mobile cells. It is made up of four separate numbers: Cell Identification (CID), Location Area Code (LAC), Mobile Network Code (MNC), and Mobile Country Code (MCC). Single cell covers several hundred meters to several kilometers. The ITU-T Recommendation E.212 defines all MCCs by the country of origin. MNC is often used in conjunction with MCC to identify unique mobile carriers/operators operating the GSM, CDMA and UMTS public land mobile networks. The LAC and CID are arbitrarily assigned by the operator. CID refers to the identity for the Base Transceiver Stations (BTS) in a particular area. Most operators masked the MCC, MNC and LAC from application developers although MCC and MNC are easily available in the public domain. Trevisani E. and Vitaletti A. [24] estimate an accuracy of 500 to 800 meters on average and recommend its use in resource discovery

service. However, accuracy can be as low as 3km in loosely populated regions. While this method does not yield a precise location estimate, its implementation is simple and comparatively fast.

An improved version of the cell identification concept comes in the form of Enhanced Cell ID (E-CID) [26]. E-CID is an improvement over CID by including Timing Advance (TA) and Network Management Records (NMR) in GSM networks. TA indicates the round trip delay time between BTS and mobile. It is used to avoid any overlapping bursts caused by multiple transmissions. Coupled with Received Signal Level, a suitable propagation model can approximate distance between a mobile and a BTS. In UMTS networks, Round Trip Time (RTT) or Reference Signal Power Budget (RSPB) is used with cell identity. RTT is the time difference between the start of a downlink frame and the reception of the corresponding uplink frame, as shown in Figure 2.5. RTT is used to calculate the distance from a Node B (similar to BTS in GSM) to a mobile. Accuracy could be up to 80m with further enhancement possible by incorporating RTT measurements from all Node Bs in the mobile active set and initiating forced handover. RSPB offers information about Node B transmission power, coverage threshold at border for a given location probability, isotropic path loss and cell radius. Radio Network Controller (RNC) controlling the Node Bs compare received power levels with power budget to obtain an estimate for the mobile's position.

Figure 2.5 RTT measurement

2.5.2 Satellites

Most smartphones today come with an inbuilt global positioning system (GPS) module. GPS [27, 28] is a wide area outdoor radio positioning system that employs orbiting satellites for autonomous global geo-spatial positioning. It determines the position of receivers using satellites signals broadcasts. The system is run by the US military and is the most utilized operational global navigation satellite system (GNSS) in service today [28].

The system consists of 24 satellites in 6 circular orbits at an altitude of approximately 20,200km with 4 evenly spaced satellites in each orbit inclined at 55 degrees. Two signals in L1 (1575.42MHz) and L2 (1227.60MHz) bands are broadcast by the satellites but only L1 is available for civilian use. Receivers make pseudo range or carrier phase measurement on L1/L2 signals to generate a location reading. A GPS receiver measures the approximate distance to each of the three or more satellites using the travel time needed for a signal to reach the receiver from the satellite. Satellite broadcasts consisting of the almanac and the ephemeris is used to calculate position using trilateration. To correct for errors, the clock bias can be calibrated to improve accuracy. Additionally, differential correction is applied to correct deliberate noise, such as selective availability, as well as variables such as ionospheric and tropospheric propagation delays. Each satellite transmits different ranging codes: Coarse/Acquisition (C/A code) and Precision Code (P-code). This mechanism allows receiver to recognize and time shift its internal clock to correspond to the code received in what is known as lock-on. When the receiver lock-on to the satellite, it can determine the exact time of signal received with respect to its own internal clock. The time taken for lock-on is known as the time-to-first-fix (TTF) and could take up to 15mins from a cold start. When the clock and the satellite's atomic clocks is synchronized, distance to each of the satellite can be calculated by subtracting the known transmission time from the calculated received time. Unlike P-code which is modulated onto both L1 and L2 carriers, C/A code is modulated onto the L1 carrier only. C/A code is less complex and of lower precision compared to the P-code, which is for military use.

Figure 2.6 GPS modulation scheme

In poor signal conditions, GPS signals suffer from multipath effects or are weakened by obstructions. The fragmentary signal would be unable to achieve a lock on until a clear signal could be received continuously. Possible sources of errors include dilution of precision, ephemeris error, ionospheric delays and satellite/receiver clock errors. Ignoring ground effects, the worse case horizontal positioning accuracy based on LOS signals is $\leq$

22m at 95% confidence interval although accuracy up to $\leq 10\text{m}$ is typically achieved. Upgrades under a GPS modernization program include improved acquisition codes to better account for ionospheric errors, radio frequency and multipath interferences [28, 29].

Assisted Global Positioning System (AGPS) [30] improves startup performance of GPS system by using additional data available from a network. These assisting information include satellite orbit & clock information, initial position & time estimate and satellite and range information. Some implementations offload processing work onto an assistance server with better computational performance and satellite signal due to its static position. It can supply GPS almanac to a receiver allowing the GPS module to lock on to the satellites more rapidly or provide precise timing information. Cell towers with assistance functions have accurate coordinates which account for various local factors affecting the GPS signal. The tower can also compare fragmentary signals received from GPS receivers with its own reading to obtain a faster location fix.

The popularity of the GPS as a leading GNSS has lead to competition from similar systems [28]. Regional satellite systems such as Beidou (China), IRNSS (India), QZSS (Japan) and also global competitors, such as GLONASS (Russia), Galileo (European Union), and COMPASS (China) are forming a new generation of GNSS, which increases the number of satellites available for positioning use. This increase can bring about better accuracy and reliability as well as improve coverage in obscured places. However, the availability of additional providers is likely to cause compatibility problems because of diverse signal broadcasts and different timings in which these new capabilities are made available. These additional features are expected to be incorporated into future mobile phones just as GPS is becoming a standard feature in smartphone today.

2.5.3 Network

Network infrastructure from service provider is utilized to determine device position and implemented without significant effects to the devices. The concentration of BTS cells affects the accuracy of network based techniques. One key challenge of network based techniques is that developers need involvement from service provider in the installation of hardware and software [31]. Such technologies employed by telecommunication companies also raise privacy concerns [32].

Uplink Time Difference of Arrival (U-TDOA) is one GSM network based solution that determines location through cellular signal. It compares the number of times a signal reaches Location Measurement Units (LMUs) which are installed together with the operator's BTSs. Mobile phones are instructed to perform asynchronous handover procedure to transmit random access bursts. TOA of random access bursts in the random access channel is measured by these LMUs. The difference in TOA measures by two different LMUs determines a hyperbola in which the mobile device may lie. U-TDOA is especially well suited for indoor and urban environments relying on multilateration to get a location fix. In UMTS networks, Observed Time Difference of Arrival (OTDOA) is used. Different Node Bs are time synchronized by LMUs or GPS receivers at the base station. Near-far problem exists for this method as serving Node B often drown out signals from distant Node Bs. Techniques like Idle Period Downlink (IDPL) can maximize the reception of distant Node Bs. Node B transmissions are synchronously paused so as to allow mobile to measure OTDOA of each neighbour Node B signal during the IDPL of serving base station. A more accurate Time-aligned IDPL (TA-IDPL) aligned idle periods of 30 μ s across the Node Bs. During this common idle period, each Node B transmits a signal that is useful for location estimation. This technique is relatively accurate for positioning purposes but adds complexity to network operation and reduces communication efficiency. OTDOA can also be coupled with cumulative virtual banking (CVB), which is based on the principle of interference cancellation. It uses virtual banking of Node B downlink signals in the software domain. Downlink signals are measured simultaneously by mobiles and neighbour Node Bs, which are transferred to a location server. It estimates OTDOA of weak Node B's signal by subtracting away interfering signals. Final position is calculated using the standard OTDOA algorithm. This method is robust in multipath environment and has the added benefit of reduced complexity and minimal impact on downlink capacity.

Enhanced Observed Time Difference (EOTD) utilizes cellular characteristics. The location is calculated by signal arrival time comparison of three or more BTSs at the mobile phone. The mobile phone records burst arrival times and recover the time difference between bursts or OTD of the reference BTS and its neighbours. This process is part of the handover procedure of mobile telephony. A position is deduced using the coordinates of the BTSs, arrival time of synchronization bursts from each BTS and timing differences between BTSs. It essentially sums the synchronization difference (Real-Time

Difference (RTD)) and the propagation time difference (Geometric Time Difference (GTD)) between two base stations.

Figure 2.7 EOTD scheme

The use of an external reference point eliminates time-synchronization but requires enhancement to phone software and additional network equipment.

2.5.4 Short Range Radio

Short range radio technologies, like Wi-Fi and Bluetooth, are now commonly available in smartphones. Wi-Fi is an Ethernet network, which requires configuration to set up shared resources, transmit files and to set up audio links. It allows mobile user to access internet through WLAN access points. Bluetooth, on the other hand, is a short range wireless personal area network (WPAN) standard often used for simple file transfer and headset connectivity. Like Wi-Fi, Bluetooth operates in the 2.4GHz radio frequency and discovers other Bluetooth devices by hopping through a sequence of channels. Bluetooth protocol does not allow device discovery through overlapping coverage from multiple devices which restricts typical service range to about 10m. Both Wi-Fi and Bluetooth technologies were not designed for location determination purposes but these network protocols were frequently adapted to implement location schemes using RSS fingerprinting. Features of Wi-Fi WLAN can be used in proximity recognition. A location can be uniquely identified by comparing and deducing radio signal patterns. These methods have proven to be an efficient way to carry out radio positioning in multipath and enclosed spaces with multiple obstructions as compared to deterministic methods. Device proximity is determined by the

similarity in Wi-Fi scanning data. Due to the nature of WLAN implementation, a few metrics can be utilized for positioning purposes such as:

- Number of common access points
- Number of access points that are not in common
- Difference in number of access points that are not in common
- Signal Strength
- Signal-to-Noise Ratio (SNR)
- Mean Square Signal Strength Deviation (MSSSD):

$$MSSSD = \frac{\sum_{i=1}^{NCAP} (P_{Ai} - P_{Bi})^2}{NCAP} \quad (28)$$

- Wireless common power percentage (WCPP):

$$WCPP = \frac{\sum_{i=1}^{NCAP} (P_{Ai} - P_{Bi})}{\sum_{i=1}^{AP_A} P_{Ai} + \sum_{i=1}^{AP_B} P_{Bi}} \times 100\% \quad (29)$$

where P_{Ai} : Signal strength of AP i detected by device A (in dB)

P_{Bi} : Signal strength of AP i detected by device B (in dB)

AP_A : Total number of APs detected by device A

AP_B : Total number of APs detected by device B

The most notable project involving short range radio signature in indoor Wi-Fi fingerprinting location schemes is RADAR [15, 33, 34]. RADAR was one of the earliest radio frequency based systems for locating and tracking user in an enclosed environment. Two approaches were proposed. An empirical method [15] that used NNSS can estimate user's location up to 2 to 3m accuracy. The main limitation was the significant effort required to construct signal strength data sets for each environment of interest (different floors, walls, amount of people etc). When base stations are relocated, the calibration process has to be re-examined. Thus, a radio propagation model [15] approach is added to provide an alternative to empirical methods. Structures inside a building are known to cause reflections, diffraction and scattering of radio waves. RADAR considered the Floor Attenuation Factor (FAF) propagation model as the most cost effective method to provision positioning services and acknowledged that accuracy is poorer than the empirical method. Further extensions to this work [33, 34] were made to address the shortcomings of the basic RADAR system. A history-based continuous user tracking algorithm models user's movement and any physical constraint due to such movement. This Viterbi algorithm-based technique minimizes the signal aliasing effects and significantly reduces estimation error by disambiguating between candidate locations. Access point based environmental profiling was applied to account for changes in

environmental factors. Radio maps for the appropriate situation are selected based on collaborative information exchanges between access points. An additional channel switching algorithm allows RADAR to operate in networks that employs frequency reuse.

It is practically unlikely to obtain perfect match between signal signatures to a stated location owing to the fact that signal propagation is fundamentally random. This deficiency can be overcome by statistical means. Nibble [35] is one such probabilistic Wi-Fi tracking service that uses Bayesian networks to infer position of a device. This standalone prototype is created by Multi-use Sensor Environment (MUSE) pervasive sensor infrastructure from UCLA Multimedia Systems Lab. The modular structure of this approach provides flexibility for the system to learn the locations even if APs are added, removed or spatially relocated. Other contexts can also be incorporated to further enhance the accuracy of predictions. Bayesian networks can also handle signal noise fluctuations due to motions in the environment by assigning probabilities to locations.

The major drawback of fingerprint methods is the need for time consuming prior calibration process for the training phase. A number of systems seek to overcome this. NearMe Proximity Server [36] project determines the proximity of devices from one another by comparing client's Wi-Fi AP and signal strengths. The server hosts a SQL database that monitors tables of active users, static resources and associated signal signatures. This process skips the geometric calibration process typical in signal fingerprint methods in favour of a location registration process that applies multi-user collaboration. Mobile client register with the server to receive a globally unique identifier. Each client tags a position with Wi-Fi signature of the user that was previously at the location. Short range and long range proximity lists are generated. Entities in short range proximity have minimum one Wi-Fi AP in common, while long range proximity objects are not within the direct reach of any single access point and only reachable by "hopping" through different access points that has overlapping coverage. While this system reduces the need for calibration, the level of accuracy is rather poor and it cannot account for differences in AP density, construction materials, and architecture layout.

2.6 Improving location sensing by integrating positioning technologies

Incorporation of multiple location sensing technologies is a central theme for most middleware research. MiddleWhere [37] is one such distributed middleware infrastructure for localization that differentiates push applications from the underlying location sensing

technologies. The system handles temporal nature of location information using a hybrid location model. Both push and pull based modes of interaction is supported using symbolic and geometric coordinates to handle region based and object based locations. This approach helps to support spatial relationships between objects. However, it can only support infrastructure based and push based LBSs. Distributed communication among middleware components, LBSs and the wrapper of the location technologies is enabled using Common Object Request Broker Architecture (CORBA).

Other middleware, such as HyLocSys [38-40] presented a software architecture that provides an application level software API to ease the inclusion of heterogeneous wireless positioning technologies in an application. A hybrid location model underlying the API is proposed to enable retrieval and storage of hybrid location data. This method which is commonly used in LBS applications is superior to using geometric location data because locations are represented in a more user oriented format. It combines the advantages of hierarchical and graph-based representations by providing interconnection and containment relations among locations. This method further enables the development of LBS for mobile devices independent of outdoor or indoor technologies by combining the use of Bluetooth and GPS location-sensing technologies. HyLocSys did not consider the use of WLAN or mobile networks as further supplement to location sensing tools.

Redpin [45-46] adopted a fingerprint-based indoor localization approach that combines GSM, Bluetooth and Wi-Fi signal readings. The system overcomes the need for calibration using an adaptive indoor localization system. It utilizes a simple distance measure and locator algorithm with the help of user collaboration. This approach trains the system while in use. The Redpin server stores fingerprint signature in a central database and allows mobile clients to retrieve maps. A symbian based sniffer component in the mobile client keeps an isolated asynchronous thread for separate signal type. This approach requires all sensors to be activated together but the time taken to obtain meaningful measurements varies for each technology. It is also easily confused in congested radio environment. The benefit is that its user collaboration scheme replaces the need for training phase. Enhancements can be made in the form of pattern recognition techniques.

Deblauwe N. & Ruppel P. [25] presented a strategy to combine GSM cell identification and GPS for LBS, which suits multi-user environment and achieve significant energy

savings by reducing GPS runtime. GPS usage is reduced by detecting if cell identification has achieved enough positioning accuracy for application requirements. This strategy reduces energy consumption and improves efficiency for mobile resources.

Other than integrating multiple technologies, there are also research into improving coverage and accuracy using crowd source techniques. Place Lab [47-50] is one such open community initiative for wide area radio beacons database to encourage the widespread adoption of location aware computing. It uses a pattern-based, layered structure to support modular development location based computing. Place Lab depends on the prior knowledge of the position of the infrastructural equipments and only supports the proximity positioning technique (accuracy is dependent on wireless beacons coverage) and handset based positioning approach. The system works by logging beacon availability recorded by users as they go about their daily routine. This allows developers to access a common facility for location tracking using simple sensors to acquire radio beacons. Interfaces provided for application include components, such as Spotters, Mappers and Trackers [49]. Spotters abstract the hardware that senses the environment. Four standards spotters are implemented: Wi-Fi, GSM, Bluetooth and GPS. Mappers are static databases utilized by trackers to get location information for spotter measurements. Wardriving activities can help to complement the databases. Trackers produce the location estimates by utilizing the stream of spotter observations and the persistent data from mappers. It is estimated the system achieve 15-20m accuracy for client devices that can detect at least 3 separate beacons during a ten seconds window period [48]. Adopting such technologies on mobile phones boast the implementation on a mass scale making it easier to create location-based applications.

LBS technologies should be supported on a variety of different mobile networks. The use of WLAN to supplement positioning is also widely used as it provides an alternative source of positioning data and is available in indoor environment where GPS is constrained. Different mobile networks also use different localization techniques. Moreover, designers also have to contemplate design issues due to international roaming. Operators at the national and international level have to cooperate to iron out issues with network support. Being able to utilize positioning services overseas is especially essential to roaming mobile clients as visitors to overseas destination have a greater need for LBS.

Thus, interoperability is a key research area to elevate LBS for mass market. An integrated multipurpose location solution is highly desirable.

The Alipes Architecture [41] describes a integrated positioning technique using GPS in outdoor situations and WaveLan positioning servers for determining indoor locations. It outlines generic positioning protocol for positioning information exchange between devices and other networked services. Extensible Markup Language (XML) scheme is used to describe positioning information which forms the core function of the positioning platform. The XML structure is replicated throughout the localization platform to represent any location information. This may allow any similar XML based middleware architecture to future proof for new and upcoming location technologies. Other researches propose appropriate abstractions in order to structure the components of a generic positioning architecture. One such proposal is a layered location stack [42] to partition layers and passes specific parameters between them. An API is then designed to integrate outdoor and indoor location sensing technologies. However, this approach does not incorporate spatial model of the physical scenario of interest.

2.7 Conclusion

This chapter introduces the different aspects of localization technologies and how such techniques had been implemented for positioning systems. There is, however a deficiency in how different wireless systems can be integrated especially with the continual emergence of new technologies. The impact of new communication technologies is uncertain and poses a great challenge to system that seeks to augment different proprietary systems. From the numerous issues regarding location context reviewed so far, little work has been done on mobile service applications distribution using location as the context. In the next chapter, we consider context modelling as a tool for improving ubiquitous computing. Such a modelling tool can help develop a structured approach to handle location information for localization purposes, which will open new ways for mobile service applications distribution.

CHAPTER 3

Adaptive Context Model for Distributed Event based Systems

To achieve the objective of ubiquitous computing, intelligent systems need to understand implicit situational information, an area typically lacking in human-computer interaction. Dey et al. [51] defined this implicit situational information as context, or “*any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical/computational object that is considered relevant to the interaction between a user and an application, including the user and application.*” Further extensions and refinements [52-54] to this context definition were commonly debated. The ability to provide ubiquitous computing services improved tremendously due to the adoption of location information as a key context to realize such services. Applications such as navigation aid, museum guide and mobile marketing, etc. have used location information to aid their service implementation. Other contexts likewise can provide opportunities for better mobile computing experience by improving context awareness. This chapter introduces a novel context delivery model that is suitable for mobile service agents. It describes the service agent processes for context data such as interpreting, processing, prioritizing and delivering context from providers through a scheme known as context refinement. This is an essential component for proximity based service agents as it defines how we make use of context to deliver relevant content to mobile users.

Definition 3.1: *Context awareness is defined as the ability to capture and adapt contexts in order to achieve an objective or a task. A service can be considered context aware if it uses context to adapt its behavior and provides task relevant information to the user. The use of context in intelligent systems does not necessary equates to context awareness. It is the ability of systems to modify behavior to make itself relevant that defines the context awareness of the system[51].*

The modelling and management of contextual information has been a hot research focus as current models lack formal system of identification and does not apply universally to

heterogeneous systems. Contextual information can be derived from multiple sources and supplied by different providers. Many context providers employ proprietary technologies that are often inoperable with others. This results in integration difficulties which brought about issues like flexibility and scalability concerns when adopted on a wider scale. Current research looks at developing multi-stage context provisioning processes to cater for different granularity needs and improving the quality of contextual information that application can use. This chapter addresses this issue of providing reliable context provisioning by adapting information from multiple context sources to create more robust context information. Proper context data management allowed better integration into user experience

An important concept in ubiquitous computing system must be the ability to deliver context information across multiple platforms. Context-aware systems require some means to communicate real world situation to computing systems by means of information modelling techniques. Previous proposed techniques for modelling context information include component-based [56], Unified Modeling Language (UML)-based [57, 58], object-oriented [59], and feedback approaches [55, 60]. While most discussions to improve context information modelling come to a consensus on the need for quality attributes, there are divided opinions as to which parameters are more important.

Gray and Salber [61] presented a way of analyzing sensed context information for context aware applications. They identified and listed coverage, resolution, accuracy, repeatability, frequency and timeliness as important attributes for information quality. Buchholz et al. [62] explained that a new concept of quality is essential for the provisioning of context aware services in an interorganizational manner. They termed this type of quality as *quality of context* (QoC).

Definition 3.2: *Quality of context is defined as “any information that describes the quality of information that is used as context information. Thus, QoC refers to information and not to the process nor the hardware component that possibly provide the information”*[62].

This includes parameters like precision, probability of correctness, trustworthiness, resolution and up-to-dateness. This view is shared by Krause et al. [63] who further concretize the notion of QoC with a refined definition.

Definition 3.2.1: *Quality of context is “any inherent information that describes context information and can be used to determine the worth of the information for a specific application. This includes information about the provisioning process the information has undergone (history, age) but not estimations about future provisioning steps it might run through”[63].*

Sheikh et al. [65] extended on this work by properly defining the attributes and quantifying the QoC indicators. Zimmer [64] identified spatial origin, age, reliability and degree of relationship as other attributes for QoC based concept for a context management system. Such a system improves context recognition rates for applications in large scale ubicomp settings.

Theodorakis et al. [66-67] introduced a context information model that utilized context refinement. The process is implemented in stages through replicated application of certain operations on the contents of a context. These actions include (1) addition of a new object, (2) specialization, generalization, or classification of an object, (3) addition of a name to an object, (4) addition of a reference to an object, and (5) application of the previous operations to the contents of a reference. It results in a context that extends information contained within the original context. Hochstatter et al. [68] implemented (as part of its context provisioning process) a context refinery to carry out context refinement for cellular networks. Its sub processes include transformation between different representation formats, extension of context information using quality attributes and merging of context information to obtain new ones. In the Adaptive & Context-Aware Services (ACAS) [69] project, context refinement was adopted using a rule-based approach unified in Tryton rule language. There are certain adverse implications over the practicality of such an approach as increasing the number of rules could potentially result in an exponential surge in processing delays. Context information refinement was also previously applied in medical systems [70] to tackle quality issues in a dynamic pervasive healthcare environment. The system’s refinement processes use categorization, normalization and extending attributes to reason and derive refined information. However, its singular aggregation approach has limited capacity to progressively upgrade context information and retrace aggregation if needed.

3.1 Dimensions of context information

Context delivery involves a variety of processes that is different from conventional computing services, which include acquisition, refinement and dissemination of context data. The circumstances pertaining to the understanding of implicit situational awareness is essential to the use of context information. Contexts derived from a situation exhibit properties that are different from contexts derived from user input or computed data.

3.1.1 Temporal Properties

Context information can be influenced by the timing and manner in which it was obtained. Most papers identified temporal properties [71-75] as an important characteristic in context information. Context information can be static or dynamic. Context that is static in nature refers to invariant aspects of the situation. For example, a mobile phone's International Mobile Equipment Identity (IMEI) number is a form of static context. It is a characteristic that remains as part of the understanding of that subject regardless of time. In contrast, dynamic context exhibits properties changes over time. Persistence of such dynamic context information is also variable. For instance, a mobile phone subscription with a telecommunication service provider is a form of long-term relationship while its association with a base station is a short-term relationship that can change abruptly as it moves from one coverage area to the next. The time element in interpreting a situation determines the understanding of the state of the entity. Data supplied directly by users are deemed to be reliable and static in nature. Sensed data on the other hand are dynamic in nature and can be rather unreliable due to uncontrollable or inbuilt factors like noise and sensor errors [76].

Current state of activity is often used to derive the latest situation but previous known facts or future plans can also be just as useful. For example, in the absence of more reliable sensors, a location service can access historical habitual records of user activity to predict current location. Similarly, the location service can look up the user's organizer to determine the planned activity at this time of the day to make an educated guess of the present position.

3.1.2 Multiple Representations

A single set of context information can take on numerous forms [71]. For example, mobile device location can be represented using physical coordinates detected by onboard GPS sensors, or as a location name like Mr. Bean's Restaurant or even as an estimated reference point like a spot 3km north of a particular mobile base station. The mentioned representations are referring to the same context but abstracted differently. Numerous factors like sensor capability, availability of information etc. affect which representation is the most appropriate under different circumstances. The cost of obtaining more complex context information also affects the representation. If a user device is interested in which room it is currently in, providing it with a room name might be sufficient instead of providing it with a more detailed GPS reading. It is essential to capture the correlation between different representations such that systems can use the data by adapting its level of abstraction.

3.1.3 Interrelated Dependencies

Context information is interrelated [70-71] and only useful to end user if it is relevant to the task concerned. Higher order information can be obtained from multiple pieces of lower order information. A piece of information can be both higher order as well as lower order at any time depending on the way it was used. It is possible to improve on the value of the information by mixing multiple sets of context information. Different context types also exhibit high degree of relationship [70]. For example, location-type context information coupled with user profile from database-type context information can be used to predict a likely activity or event. The ability to capture wide spectrum of relationships between events is useful in correlating information to clarify the situation.

3.1.4 Imperfect Information

Ubiquitous computing environments are highly dynamic attributed to a variety of reasons. Unlike user input, sensed context are not as reliable because they suffered from discrepancy problems. Context information can be described as imperfect [53, 71-72] as they can be incorrect, inconsistent, incomplete or a combination of them. Incorrect context arises when the information failed to reflect the true state of what it is modelling thereby distorting the overall interpretation of the scenario. Inconsistent context shows contradictory information when compared against other sources leading to doubts over the

trustworthiness of the information. For incomplete context, some aspects of the context are unknown due to masking of data.

Context imperfection could be a result of differences in processing delays between the initial production of context to its eventual consumption by end users. This leads to outdated information that may contradict with information from other sources. This mix up leads to complication for future processes. Another reason could also be the differences in quality of sensor devices.

3.2 Context Service Delivery

The concept of context delivery works on a multi-layered context delivery architecture as depicted in Figure 3.1. Each stage has its unique role in the propagation of context information. The context provisioning scheme for heterogeneous sources includes processes such as collection, handling, management, tracking and dissemination. It also involves the decision sub processes required to refine context information using QoC parameters. The process is adaptable to changes in QoC and offers applications better and more adaptive context information. In less ideal conditions, the process utilizes poorer component context information whenever possible. The key problem here is to provide a mechanism to differentiate events such that only significant ones are captured and unwarranted event triggers are minimized.

Figure 3.1 Context Delivery Framework

The first level in the context service delivery process is the *acquisition layer* which handles context discovery and retrieval. This layer consists of context providers that generate digital data from observable contexts in the environment and other recorded contextual information sources. Some context providers are sensor type systems while others provide contextual information using databases. Multiple possible context sources offer information with variable levels of QoC despite them referring to the exact same

situation. This layer forms a collection level where context information is captured without prioritizing its importance to the user.

The *identification layer* determines relationship between context providers. In this layer, a context management service (CMS) detects and applies updates to context information as and when an update is available from information providers. CMS provides the subscriptions for all available entities in the acquisition layer and maintain soft states to track for particular outcomes.

The *transformation layer* handles associations between context information and applications. It does so by aggregating available information using context refinement. All context information undergoes a context refinement process to determine if the context information can be improved. Such enhancements make context information more relevant through a set of established QoC parameters. Context refinement is further elaborated in Section 3.3. The intent of this layer is to cut down duplication of information and improve situation awareness by closing information gaps between context sources. The context aware service (CAS) in this layer specifies appropriate entities, roles, and relations to operate user activities. CAS acts as a brokering service between CMS and the context end users. Context end users here refer to applications that request for context information. CAS observes for changes in the situation and anticipates application needs for various context forms. Dissemination and delivery of the refined context information are also its responsibilities. Context users passed context information requests to the CAS which creates soft states to track for a particular outcome. The required information is delivered to the end user through the CAS monitoring the event and triggers an alert whenever a relevant one is initiated.

The *application layer* at the end of the context delivery chain consists of context consumers. During application runtime, context users determine the contexts required and inform CAS. When relevant context information is available in the transformation layer, CAS notifies the application which can in turn modifies its behavior and adapts its content in response to the context information provided.

The entire context provisioning process shows the flow of context information from acquisition to its use in application. Such architecture will allow software developers to concentrate on application logic and leaves context acquiring to a middleware platform.

3.3 Context Refinement

Definition 3.3: *Context information refinement is defined as an improvement transformation to progress information using available contexts. The resulting product can be used by context users or acts as an intermediary for further improvement.*

Context information is represented as a computational object qualified with QoC parameters. Previous QoC researches [61-65] were considered to determine a set of objective parameters where context objects can use to compare between information.

3.3.1 QoC Parameters

Context *integrity* refers to the believability and trustworthiness of information from a context source. Providers that contribute high integrity information has high degree of trust. High integrity sources tend to also exhibit strong values in other QoC parameters. This allows system to make comparison quickly and channel the important ones as priority downstream. *Timeliness* is related to the age of the information. The latest information is often more relevant compared to older ones. In most cases, older contexts can be discarded without any significant consequences. Timeliness is quantified by referencing the timestamp against current time. The dynamic nature of this parameter is a characteristic of all real time applications. As such, the comparison scheme has to regularly refresh itself to reflect changes to this parameter as it gradually becomes outdated. *Reliability* is the probability of correctness in the data provided by the context source. This parameter refers to the dependability of the source to provide error free data that are within its operating specification. Failure to provide accurate data regularly signifies inherent faults in the sensor system and should be depended on as little as possible. *Precision* is defined as the granularity of information corresponding to the sensitivity of the context provider. For example, a digital thermometer has higher precision than mercury-in-glass thermometer. In most cases, precision of individual sensor might be poorer as it could sense only a portion of the overall situation. It is therefore possible to achieve better precision by integrating readings from multiple sensors.

Each attribute is rated on a scale of 1 to 5 according to Table 3.1. Context information is assessed by its QoC parameter tags when received. Context information that has already been classified might be updated to reflect changes. For example, an information object with timeliness rating '4' might drop to rating '2' after a few hours because the data is old

and no longer useful. Table 3.2 shows an example of context information tagged with QoC attributes and arrange in order of decreasing QoC levels.

Table 3.1 Rating description for QoC parameter tags

Parameter	Poor		Average		Strong
	1	2	3	4	5
<i>Integrity</i>	Non trusted	Believable	Neutral	Trusted (external)	Trusted (internal)
<i>Timeliness</i>	Outdated	Old	Still Relevant	Latest	Current
<i>Reliability</i>	Error Prone	Some Errors	Manageable errors	Negligible errors	Near Perfect
<i>Precision</i>	Poor	Vague	Accurate	Sensitive	Precise

*A rating '0' is used as an error control mechanism to represent parameter that are indeterminable due to any unexpected errors.

Table 3.2 QoC tags for context information

<i>QoC rank</i>	<i>Context ID</i>	<i>Integrity</i>	<i>Timeliness</i>	<i>Reliability</i>	<i>Precision</i>
1	Context 5	5	4	3	3
2	Context 3	5	2	4	5
3*	Context 4	4	3	3	3
4*	Context 1	4	3	3	3
5	Context 2	3	4	1	2

*Note: Context 1 and Context 4 exhibits similar parameters. Context 4 is ranked higher because it was assessed later.

When comparing between context information, a 'medal tally' system is used. Context information is ranked higher on the QoC scale if it has a higher integrity value compared to another. If the comparison at the integrity level is equal, the next attribute is used. This comparison continues until all the attributes are exhausted. The QoC scale forms the basis of comparisons in context refinement sub processes.

3.3.2 Sub Processes

In this context refinement architecture, all context information is modelled as a multi-tier context object (CO), as shown in Figure 3.2.

Figure 3.2 Modelling multi-tier context objects (COs)

Tier 1 COs contain context information direct from context providers (CPs) and tagged with source QoC attributes. COs in downstream tiers consist of at least one CO from tiers directly before it. Downstream COs can contain different QoC attributes from its composition COs. This multi-layer concept to differentiate COs helps to trace the composition COs and offer a choice of utilizing lower tier CO if required either voluntarily or due to circumstances (e.g. during downtime of a component context provider). The context acquisition layer collects data from context providers and delivers them to the CMS for tracking purposes. Context information is aggregated in the CAS by performing systematic reasoning. The process is designed to allow CO to progressively upgrade quality levels when possible, while offering the use of lower quality information as necessary. Sub processes include combination/extrapolation, deduction, interpolation, selection and filtration.

Combination/Extrapolation: Each sensor source acquires its readings independently and has its own interpretation of a situation. Sensor based context providers can be in the form of wireless signals detector, measuring instrument etc. Due to structural differences, each context source provides CO with different QoC levels. Improvement in QoC levels can be achieved by integrating COs. This results in the production of CO that may have greater QoC levels than its constituent COs. This process propagates along the context refinement process chain until the desired QoC level is achieved or when it is impossible to improve any further as seen from Figure 3.3.

Figure 3.3 Combination/Extrapolation

Deduction: Prior knowledge of a situation is available and applied in this sub process. This knowledge could be in the form of stored historical contexts, predictive contexts, user input contexts and contexts from external system etc. They provide supplemental information which coupled with current contexts can potentially improve the

understanding of current user situation. Some examples include calibration data stored in memory, server records, internet resources etc. COs supplied by databases can provide additional information to make quick estimation in the absence of relevant current data from sensors as seen in Figure 3.4 .

Figure 3.4 Deduction

Selection: Context information is imperfect resulting in some derived contexts to be inaccurate. In presence of these erroneous sources, combination or deduction of COs is not the most appropriate action. The erroneous context information should be sieved out when detected to avoid corrupting the decision process. It is desirable to remove erroneous sources at the earliest possible stage to minimize data traffic at the application level. Selection happens when an undesirable downgrade in context information at the higher tier level is experienced after combination. This is detected by a significant drop in QoC levels compared to the constituent CO. The selection process halts the combination attempt and drops higher tier CO in favor of the lower tier CO as seen in Figure 3.5.

Figure 3.5 Selection

Interpolation: Sometimes, unrelated context information might provide useful information for COs further downstream. For example, two tier 1 COs produce an intermediary tier 2 CO, which has lower QoC than one of its constituents. This tier 2 CO however, provides

for an information gap to produce a tier 3 CO with much higher QoC levels. As such, it is not eliminated by the selection process and retained in anticipation for future use. It would otherwise be impossible otherwise to obtain the tier 3 information without the intermediate tier 2 CO as illustrated in Figure 3.6.

Figure 3.6 Interpolation

Filtration: In multi-sensor environment, some context sources offer more comprehensive data coverage. The poorer source can be a subset of the better source. Combination of the two sources results in redundancy as the higher tier CO created takes on the QoC of the better source without any improvements. The CO is eliminated upon creation. Filtration helps to eliminate poorer contexts which do not provide additional information as seen in Figure 3.7.

Figure 3.7 Filtration

2.3.3 Concepts and Implementation

The approach to context delivery is to offer refinement of context information through the use of QoC parameters tagged to information objects. The timeliness parameter monitors the time elapsed in COs thereby supporting the temporal characteristics of context information delivery. The framework allows multiple representations of context information because the transformation taken in the refinement process can translate

information into different forms depending on application needs. Dependencies between different COs are supported as the relationship are tracked and maintained in the refinement process. The ranking system based on QoC helps to manage information that may not be fully accurate by allowing leeway for poorer data to interact with other information and rapidly improve themselves whenever possible, even in cases of errors in certain parts of the information. Such a context refinement model supports the characteristics of most context information.

The context model is illustrated using a computer terminal allocator service as an example. The allocator service uses infrastructural Wi-Fi access points as primary context providers and stored data in the mobile handset databases as supplementary context providers. A laboratory is divided into three separate areas for different computing classes. The allocator application hosted on a user's mobile scans the environment for Wi-Fi signals and establishes if it is in the vicinity of the laboratory by comparing the observed signal readings against signal signatures from previous calibration. The application communicates with the lab server to request a seat allocation through a Wi-Fi connection. Server checks the identity to establish which class the user belongs to and allocates the next available seats in his class. This information is then relayed to the user which displays the classroom and the allocated seat. Such an implementation can also be applied to airline and theatre seat arrangement. In this scenario, context information received by the user's mobile enters a series of transformation to implement the allocator application. The calibration data in the device memory provides CO 'X' and the observed signal strength by the inbuilt Wi-Fi interface provides CO 'Y'. The context refinement of CO 'X' and 'Y' generates a location alert denoted by CO 'Z' using a deduction context refinement sub process. The new context information in CO 'Z' informs the application that the user is in the vicinity of a service provider (lab server). Their respective QoC attributes and the effect of applying context refinement deduction sub process are stated in Table 3.3.

Table 3.3 Deduction process effect on CO ranking

<i>QoC Rank</i>	<i>Context ID</i>	<i>Integrity</i>	<i>Timeliness</i>	<i>Reliability</i>	<i>Precision</i>
Before Deduction					
1	X	5	2	4	4
2	Y	4	5	3	3
After Deduction					
1	Z	5	4	3	3
2	X	5	2	4	4
3	Y	4	4	3	3

From the observation, we could see the deduction process improves QoC levels of the generated CO ‘Z’. More detailed information is known about the current situation due to the two context sources. This is reflected by the higher QoC levels in the CO ‘Z’. Note that CO ‘Y’ timeliness attribute has drop between before and after deduction to reflect time progression.

The CAS context refinement scheme is designed in a way that developers can trifle with parameters rankings and add any additional parameters to offer alternate versions for different application classes. The order and number of parameters, which are shown in the allocator service example, are not generic to every application. This implementation only seeks to serve as a guide that adds higher weight to trusted context providers. Alternate CAS can incorporate other QoC parameters or choose another parameter to be set as a higher priority. For example, a temperature monitoring application can assign precision to be of a higher priority than the timeliness parameter.

3.4 Conclusion

Context modelling is an important aspect in localization as it dictates the flow of information input into a system and the corresponding information outflow. The context model describes in this chapter is suitable for different form of contexts. Most location based mobile services require different granularity of accuracy. The quality of location data collected affects its usability. Some applications might be satisfied with the name but others might need detailed coordinates. Thus, the design of a context delivery model for location context has to account for difficulties associated with mobile services. Loss of connectivity is normal and connected session tends to be patchy. A key for the location context delivery model is how to react to frequent unannounced changes to network bandwidth and resource availability. The model must support user autonomy and control to enhance the overall pervasive computing experience. Localization techniques differ in terms of accuracy and the manner in which they operate. The next chapter describes the practical consideration on how the context refinement scheme can be applied to mobile localization based on the environment in which it operates.

CHAPTER 4

Wireless Localization for Distributed Mobile Environment

This chapter describes the technical limitations in mobile localization and its potential impact on proximity based application distribution. Utilizing timing as a distance tool requires good LOS propagation paths from the wireless source. Radio propagation models that used signal strength are ineffective in accounting for unpredictable variation like fading, multipath, interferences, etc. as signal strength does not correlate directly to distance, especially in cases where heterogeneous transmitters are used. Performance gain in areas with high density reference nodes is also limited. Signal pattern mapping techniques require substantial effort in generating signal signatures and suffer from signal drag effect [77-78]. When a mobile is in motion, the received pattern will contain more values from base stations positioned opposite of the moving direction than signals from base stations positioned in the forward direction. Lastly, directionality based methods are ineffective in indoor spaces as multipath effects obscure the true direction of the transmission. The additional cost involved in installing directional antennas is prohibitive for mobile applications. Such antennas can only be placed at the reference sites and may not scale well with large number of mobile entities.

4.1 Consideration Factors in Mobile Environment

4.1.1 Noise

Noise characteristic of wireless channels is a well known phenomenon. A common source of noise in wireless channels is white noise, a noise that disrupts an input such that the output becomes additional function of the input noise. White noise [79] can be assumed to be a random process $z(t)$ with a mean 0 and variance $\rightarrow \infty$, such that $z(t)$ and $z(s)$ are independent for $t \neq s$ which result in:

$$E(\int f(t)z(t)dt)^2 = \int f(t)^2 dt \quad (30)$$

Thus, $z(t)$ can be seen as a continuous analogue of an independent but identically distributed random sequence. At each time t , $z(t)$ has infinite fluctuation. White noise is often generalized to be a Gaussian process [79]. A stationary Gaussian process X_t is determined by

$$(1) E(X_t) = m \text{ (independent of } t)$$

$$(2) E(X_{t+u} - m)(X_u - m) = \theta(t) \text{ (independent of } u)$$

The function $\theta(t)$ is positive definite and $\theta(0) = E(X_u - m)^2 \equiv \sigma^2$. Thus, assuming that $\theta(t)$ is continuous, then by the Bochner [79] theorem,

$$\theta(t) = \int e^{it\lambda} f(\lambda) d\lambda. \quad (31)$$

Here $f(\lambda)$ is called the spectral density of X_t . Note that

$$\int f(\lambda) d\lambda = \theta(0) = \sigma^2 \neq 0. \quad (32)$$

Informally, white noise is regarded to be stationary Gaussian process, such that the spectral density function is constant. Thus, the variance $\sigma^2 = \infty$. Such a process does not exist as above since the function $\theta(t)$ is given by the Dirac function $\theta(t) = \delta_0(t)$ at the origin. In wireless communication, this form of noise is described as Additive White Gaussian Noise (AWGN). Gaussian noise originates from many natural sources making AWGN a good model for satellite and deep space communication links but not for most terrestrial links as factors such as multipath effects, terrain blocking, interference ground clutter and self interference take precedence in modern radio systems operations. Due to its negligible effects on overall net noise received, AWGN would not have significant impact on mobile wireless localization operation in a proximity based distribution system.

4.1.2 Channel Interference

Co-channel interference (CCI) is a form of cross-talk between two different radio transmitters reusing old frequency channel [80]. It occurs when two or more simultaneous transmissions happened to be on the same channel. This form of interference is inherent in any type of system that employs frequency reuse methodology. They can also be caused by adverse weather condition, poor frequency planning or overcrowded radio spectrum. During weathers of abnormally high atmospheric pressure, signals that normally exit Earth, pass through the atmosphere and are reflected by the troposphere instead. This

causes the signal to rebound back to Earth at another location, causing unexpected interference to transmitters in the vicinity. Poor planning of frequencies can cause CCI due to duplicate frequencies being unintentionally used. In urban areas where demand for the radio spectrum is more intense, tightly spaced channels may result in aliasing. Co-channel interference can be controlled by good radio resource management scheme.

The calculation of the co-channel interference power is critical for system designers. Overall carrier-to-noise ratio (CNR), which includes interference components can be calculated by:

$$(C/N)_0 = \frac{1}{\left[\frac{1}{(C/N)_{UL}} + \frac{1}{(C/N)_{DL}} + \frac{1}{(C/I)_0} \right]}, \quad (33)$$

where the subscripts, UL and DL , represent uplink and downlink, respectively. Carrier to interference ratio $(C/I)_0$ is derived by [81]:

$$(C/I)_0 = \frac{1}{\left[\frac{1}{(C/I)_{CC}} + \frac{1}{(C/I)_{IM}} \right]}, \quad (34)$$

where the subscripts CC and IM indicate co-channel and intermodulation interferences respectively. $(C/I)_0$ calculated here may also include other types of interferences. $(C/I)_0$ for co-channel interference only is calculated by:

$$(C/I)_0 = \frac{1}{\left[\frac{1}{(C/I)_{CC-UL}} + \frac{1}{(C/I)_{CC-DL}} \right]}. \quad (35)$$

$(C/I)_{CC-DL}$ (Downlink co-channel carrier-to-interference ratio) is a function of the reuse number and the aggregate power due to the sidelobes of interfering co-channel. $(C/I)_{CC-UL}$ (Uplink co-channel carrier-to-interference ratio), however, is dependent on reuse number and the amount of co-channel users that are transmitting simultaneously and receiving at the sidelobes of the interfered beam. Channel interference can also be in the form of adjacent channel interference (ACI). This interference is caused by extraneous power from a signal in an adjacent channel [82]. They can be caused by insufficient filtering. For example there could be incomplete filtering of unwanted modulation products in frequency modulation (FM) systems, improper tuning or poor frequency control in either the reference channel or interfering channel or both at the same time.

Frequency spectrum is a precious resource in cellular mobile communication. Cellular frequencies are separated into non-overlapping spectrum bands assigned to different cells. However, frequency bands can be reused as the exact spectrum bands can be reassigned to more distant cells. Co-channel interference can arise as a result of this reuse of frequency. This crosstalk effect affects signal reaching the receiver, which in turn affects wireless localization performance. For WLAN systems, the 802.11 network standards transmit at 2.4 GHz or 5 GHz. Some of these standards use orthogonal frequency-division multiplexing (OFDM), which is a more efficient coding method that splits that radio signal into several sub-signals before they reach a receiver which reduces interference. Wi-Fi radios can "frequency hop" between the different bands to reduce interference and allow multiple devices use the same wireless connection simultaneously. Such multiplexing also helps to reduce the impact of crosstalk on wireless localization.

4.1.3 Multipath Effects

Travel time measurements for determining the distance between transmitters and receivers generally require LOS propagation to generate accurate results. Multipath is used to describe the multiple paths a radio wave may propagate from the transmitter to the receiver. In a complicated environment, multipath fading comes into effect when an antenna receives a transmitted signal that is the sum of the desired LOS signal plus one or more Non-Line-of-Sight (NLOS) signals [81]. NLOS signals are caused by reflections or diffraction off structures and obstacles. Radio waves that are received in-phase create constructive interference resulting in a stronger signal received at the receiver. Those that are received out-of-phase interfere destructively and produce a weak or fading signal in the process. The small alterations in the transmission path can cause changes in the phase relationship of the two signals, producing periodic fading. Radio measurements are often biased by multipath propagations that include LOS and NLOS propagations in random proportions.

A signal propagation model can involve empirical mathematical formula in predicting the propagation behaviour of related signals in a set constraint. Such models usually predict path loss along a route or the effective coverage area of a transmitter by characterizing radio wave propagation as a function of distance and other conditions. Mixture of LOS and NLOS signal propagations cause signal strength distortion that varies from expected value estimated by signal propagation models. Common causes of interference include:

steel bar in concrete wall, metal frame, glass wall, metal door, metal cabinet and other wireless equipment operating in the 2.4GHz (ISM) radio wave range.

Location systems must account for multipath phenomenon in order to offer more accurate estimates. Multipath fading can be minimized by using space and frequency diversification [81]. In space diversity, two or more receiving antennas are spaced some distance apart. The amount of fading experienced by these receiving antennas will not be equal and usually, enough throughputs are available at one of the receiving antennas to provide a useful signal. In frequency diversity, two sets of transmitters and receivers are used with each pair communicating at a different frequency. Both sets carry the same information and are transmitted simultaneously over both frequencies. At least one of the two receivers will produce a working signal for use. The filtering of unwanted signals travelling different paths makes it extremely tedious to implement on a location system. Even if NLOS propagations can be eliminated, there are other impairments to LOS propagation caused by man-made and natural obstacles to account for. Objects not in direct LOS can also cause diffraction effects that interfere with radio transmissions. In some cases, LOS signal get attenuated by transmission reflection from the ground. All these factors had to be account for in a signal propagation model. This is often tedious and may not handle complex environment. Stochastic methods like Rayleigh and Rician distribution could also serve as multipath models for radio transmission signals in urban environments. They account for random fading of signal strength magnitude when a signal transmission passed through a medium. Rayleigh fading model is suitable for environment where there is no dominant LOS propagation between the transmitter and receiver, while Rician fading model handles situation where dominant LOS transmission partially cancel a radio signal with itself. RSS varies by the amount of LOS and NLOS paths in the environment. Using deterministic methods of wireless localization is especially difficult given the random nature of multipath interference in complicated spaces.

4.1.4 Signal Drag

Signal drag has an adverse impact on mobile localization even though it is not a major issue for wireless data traffic control. The effect is because of wireless receiver taking into account delayed RSS values in signal signature collection from outdated time frames during receiver movement. These cached signal values did not match the measurements anticipated at the location. During the movement of a receiver, the received signal

signature contains more signals from transmitters positioned opposite the moving direction than those transmitters in the moving direction [77]. These nodes could be base transceivers in the case of cellular networks or wireless access points in Wi-Fi networks. The relative positions of these nodes along the receiver movement path are important factors in the determining the presence of signal drag.

Signal drag on Code Division Multiple Access (CDMA) networks and its consequences on wardriving activities were previously investigated in [77]. Wardriving here refers to the process of using a receiver to log a trace of network availability while driving. Whenever a mobile moves within a cellular network, it constantly communicates with nearby base stations. An updated RSS information list is maintained to make handoff decisions by categorizing identified base station entries into active set, candidate set, neighbour set and other remaining set. The active set contains service base stations which a mobile phone currently uses. Candidate set contains base stations not belonging to the active set but have signals with sufficient strength to demodulate the signal. Neighbour set contains base stations likely to become candidate set for idle handover as specified in a control or paging channel message from the service base station. The remaining set contains all other base stations. Mobile handsets often compose of NMR that contains old RSS entries, which no longer corresponds to the current location. Old cached RSS values in the neighbour and remaining sets contributes the most to signal drag in cellular networks. Other causes include lengthy interval between signal updates, operational characteristics of receiver or receiver going into idle mode to save power by waking up only periodically.

Wi-Fi networks also exhibit similar signal drag. Such wireless localization systems collects signals from wardriving activities to determine approximate access points (APs) positions. The same model is also utilized to determine estimated user device location. These algorithms usually require a training phase to measure the WLAN signals in an area. This scan captures data like MAC ID and RSS from any of access points within range and a separate recording device such as a GPS records the physical coordinates. To determine the true signal strength mean for each observable access points at the location of interest, samples at regular interval in a fixed period need to be collected. The signal information recorded previously is used as a training dataset. When the training is completed, a signal signature map of the area can be generated. RSS based localization make use of this data to match possible locations. WLAN signals appear irregular even though the measurement was done at a static position as the signal propagation can be affected by factors such as

dead spots, noise interference and multipath effects. The receiver's orientation can affect signal reading as illustrated in a signal profile as seen in Figure 4.1. Obvious correlation between signal strength and the orientation of device is observed where variations of between -3 to -4 dBm of signal strength difference among different orientations as can be seen in Figure 4.2. This suggests that different orientations should be treated as distinct logical processing in a pattern based positioning system. When movement of terminal is considered, the signal drag caused by including outdated receiver signal will also create some differences to previous calibrations.

Figure 4.1 Signal strength distribution profile of WLAN access points at a static location. This information is used to create signal fingerprint signature in RSS based localization to map physical locations in known space.

Figure 4.2 Effect of terminal orientation on signal reception. Signal distributions for each of the four different orientations at a static point are collected using 300 samples at two second interval for a single access point.

Signal Dragging Error Metric (SDEM) was previously used in [78] to predict the presence of signal drag in wireless networks. It is derived from angular difference between vector sum of signals collected and moving direction of receiver. Average angular differences for every position is compared. Whenever the mean is greatly lower than 90 degrees, it means that the receiver would contain a greater amount of signals from backward nodes compared to signals from forward nodes. A value of ≈ 90 degrees implies minimal dependency between receiver movement and geometric distribution of nodes. This indicates the possibility of signal drag effects. The same observation applies similarly to all types of wireless networks.

The relation between signal drag and speed is determined by calculating the correlation coefficient between moving speed and inverse of SDEM. Movement speed was observed to correlate with amount of signal drag. Theoretically the correlation coefficient should be between the range of -1 to +1. When movement speed increases, effect becomes more apparent. Signal drag continues its effect and ceases only when estimated direction arrows matches the true moving direction when during straight motion. When the receiver changes direction, the estimation arrows will slowly converged to the new direction after a finite delay. Arrows appear “laggy” as it takes time and certain amount of distance for it to catch up to the true direction. Thus, if receiver changes direction regularly, the SDEM will increase due to the change in angle resulting in less noticeable signal drag. This is a result of receiver having more chances to catch up to the true direction.

Environmental and arrangement of the service nodes within the space are main factors causing signal drag. Signal drag is more prominent in environment where service nodes are distributed uniformly throughout the space as compared to uneven arrangement. This is counterintuitive to conventional wireless network planning where service nodes is spread evenly to maximize coverage. The sum vector will only point towards region of higher density service nodes and because of this factor, the signal drag effect experienced at the boundary region of WLAN coverage is quite apparent. Direction vectors are calculated from points in similar direction regardless whether the mobile is moving forward or in the reverse direction of movement. In conclusion, a signal drag problem will only become a concern inside a WLAN field where access points are likely to be placed more uniformly compared to boundary regions. As each access point face the movement path heterogeneously, a few access points may be susceptible to signal drag effects as compared to other access points.

To illustrate signal drag in WLAN network and its impact on wireless localization, signal data from numerous access points on a prescribed rectangular path was collected. Three time intervals and two directions were planned in this experiment. The initial direction is designated as forward direction and the opposite direction was treated as the reverse direction. Radio maps containing MAC addresses and signal strength information of access points in interest area was obtained using Ekahau HeatMapper [83]. The results of the survey determine approximate locations of access points in range. Other than collecting on-the-move signal information, signals detected at various fixed points along the path were independently collected. This is known as the static steady state path. It is used as the comparison for all other paths that has motion. Data collected from wardriving were compared with data collected when receiver is stationary. The signal strength readings were taken once every two seconds during the movement phases.

Figure 4.3 Signal readings (AP A) along the wardriving path in a 1/2 minute lap

Figure 4.4 Signal readings (AP B) along the wardriving path in a 1/2 minute lap

Figure 4.5 Signal readings (AP C) along the wardriving path in a 1/2 minute lap

Figure 4.6 Signal readings (AP A) along the wardriving path in a 1 minute lap

Figure 4.7 Signal readings (AP B) along the wardriving path in a 1 minute lap

Figure 4.8 Signal readings (AP C) along the wardriving path in a 1 minute lap

Figure 4.9 Signal readings (AP A) along the wardriving path in a 2 minute lap

Figure 4.10 Signal readings (AP B) along the wardriving path in a 2 minute lap

Figure 4.11 Signal readings (AP C) along the wardriving path in a 2 minute lap

Figure 4.12 Estimated locations of access points A, B and C (top to bottom). The circle indicates the region where most influence is exerted along the movement path due to the AP's proximity. The sum vector indication from a single AP with respect to the actual ground motion is shown using the arrows. Overall net signal drag effects depends on the combined vectors from all visible APs.

Figures 4.3 to 4.11 show the comparisons of signal information both moving and stationary recovered from each AP. The signal strength of each AP is normally stronger and steadier when the receiver is in a stationary position compared to a moving receiver. Many APs were undetected during certain period of time in moving receivers. Experiments carried out over three separate time frames is used to find out the effect of speed on WLAN positioning. A static path is used as the baseline against movement paths with variable speeds. The path used by receiver stayed constant through the experiments. The speed was changed so that the same path can be completed in 2-minute, 1-minute and 30-second periods. Signal strength variations observed in the forward direction is often a mirror image of the reverse direction independent of the actual speed used. These meant that signal drag is more prominent at higher speed. Some experiments see the target having been more prone to signal drag effects compare to others because the access points faced differently as evident in the data from access points A and C. Moving receiver is thus likely to be affected by access points located in the direct forward/backward of the moving path than access points placed to the side. Thus, receiver movement towards or away from general direction of previously known access points must consider the effect from these APs during the movement phase.

4.2 Mitigation Strategies

Differences in perceived signal strength during movement might not be correctly accounted for when compared with the prior calibration datasets as these were carried out under a different set of conditions during wardriving. This effect adds further complications for pattern recognition systems that utilized statistical similarities for output results. Localization systems need to account for such differences in signal strength between steady state and movements at different speeds. Thus, an active radio map calibration scheme is desirable.

4.2.1 Previous Approaches

To negate signal drag, [78] proposes distinguishing outdated or unreachable BSs from active ones in the NMRs by using three proposed methods. These methods are GT-based filtering, WCL-based filtering and RSS-based filtering. In GT-based filtering, the system treats BS data as cached whenever a BS's position is far from the mobile terminal's ground truth (GT) or actual position. It requires the use of another location service

provider to supplement the actual location with the wardriving records. A terminal's location is calculated by weighted centroid localization (WCL) algorithm using weighted average of BS positions based on RSS values. A new reference WCL location is recomputed by iteratively removing the farthest BSs from WCL results. RSS-based filtering eliminates weakest BSs consecutively based on RSS values. The result is the filtering of cached outdated BS data. In their work, all three filtering methods exhibits noticeable signal drag reduction in Wideband Code Division Multiple Access (WCDMA) and GSM networks. However, the methods do not work as effectively for Wi-Fi networks.

4.2.2 Adaptive Cluster Selection

RSS signature based location systems are generally divided into server based and client based architectures. Server based systems are typically more resource friendly but not as good in privacy controls compared to client based solutions. Server based location solution collects signal strength signatures observed by client devices to decide their approximate positions. Conventional positioning theories are limited when dealing with moving receivers. The quality of location estimate is impacted by amount of receiver movement and direction taken. Other factors that affect the magnitude of signal drag including arrangement of reference points, operational characteristic of receiver, interval between signal pattern updates and differences in idle mode operation. Terminal orientation along projected path also adds complication to the effect.

Mitigating strategy is essential to counter signal drag so as to improve localization estimates. The orientation of receiver gives a sense of the general direction of APs which can assist in determining the onset of possible signal drag. Thus, when estimating location, the system can assign a lower weight consideration to high risk service APs. Systems applying similar non deterministic methods for localization will find this useful. A cluster selection scheme handles radio map selection in wireless localization. The scheme utilized an enhanced k -Nearest Neighbors (k NN) pattern matching algorithm and backpropagation (BP) neural network classification to determine a relevant set of radio maps for the correct signal drag profile. Its effectiveness is evaluated using actual WLAN data and experiments show significant improvement in successful cluster selection. Higher success rate for grid selection reduces computation time thereby enabling better wireless localization performance. The implementation is divided into three distinct phases: collection/training phase, online calibration phase and adaptive localization phase as shown in Figure 4.13.

Figure 4.13 Mapping location estimates using fingerprint schemes and adaptive radio map selection. Location estimates are provided by sensors and trained data sets.

During the collection/training phase, RSS signatures are collected from the area of interest during wardriving to construct radio maps. RSS based localization follows the IEEE 802.11 specification processes for APs detection. WLAN receiver operates in the 2.4/5.0 GHz radio frequency spectrum serving as the communication medium between transceivers. A receiver scan probes the Open Systems Interconnection (OSI) physical layer in the communication medium and attempts to decode any relevant media access control (MAC) payload addresses in the corresponding data link layer. The outcome of MAC decoding from the physical layer depends on the received power of the transmission.

Some detected APs are not always visible throughout the entire scan. Signals from such rogue wireless APs are often weak or appear intermittently. To reduce the burden of location estimation on host devices, a smaller subset from all detectable APs is used in the collection/training phase instead so as to reduce the amount of operations. A key task here would be to determine whether signal from a prospective AP is worth retaining for further processing. Any undecodable signals from nearby APs in the communication medium are treated as busy or undetectable. Irrelevant rogue APs can be removed from the detected APs list in the input dataset by applying a noise filter. It sets a time interval t to record RSS values and APs detected. APs present most frequently within this time interval ($\geq 80\%$ of the interval) are chosen. Time t is as short as possible to reduce the time spent scanning while ensuring that sufficient information and good RSS measurements are captured for future location estimation. Previous trials revealed an optimal $17s \leq t \leq 20s$ in

a typical WLAN zone with an average of 4-8 visible APs within the coverage zone. Note that this is the typical steady state static measurement parameters for the calibration/training phase. A typical training data entry set consists of signal strengths from all observable APs and the position, i where the readings are taken:

$$(AP_1^i, AP_2^i, AP_3^i, \dots, AP_D^i, x^i, y^i) \quad (36)$$

The sensing and subsequent indexing of AP signals can be classified into a matrix:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_L \end{pmatrix}_L = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1D} \\ a_{21} & a_{22} & \cdots & a_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{L1} & a_{L2} & \cdots & a_{LD} \end{pmatrix}_{L \times D} \begin{pmatrix} x_1 \\ \vdots \\ x_D \end{pmatrix}_D, \quad (37)$$

where

Vector $\mathbf{Y} = [y_1, y_2, \dots, y_L]^T \in \mathbb{R}^L$ represents measurements from selected APs,
Matrix $\mathbf{A} \in \mathbb{R}^{L \times D}$, $L \leq D$ contains binary weightings for APs selections,
Vector $\mathbf{X} = [x_1, x_2, \dots, x_D]^T \in \mathbb{R}^D$ represents the signal measurements from all available D detected APs.

The next criterion is to determine the number of relevant APs, n such that there are sufficient distinguishable APs to determine a position in physical space. An area of interest is divided into equal size location grids. Condition for n is such that all location grids are covered by at least n number of APs. This ensures a minimum set of APs to distinguish positions with accuracy similar to the length of location grid edge. To minimize data search complexity, a unique radio map clustering module is used to reduce search space required to return a result. It categorizes locations that shared the same set of APs. Each group of such locations is a cluster. During online calibration and adaptive localization phase, the module only searches the cluster that contains detected APs instead of probing the entire radio map search space for a match.

Figure 4.14 Three-layer backpropagation neural network implementation for NNSS pattern recognition and classification using k NN. First layer handles input distribution. Second layer provides feedback control. Last layer forms neurons for output classifications.

The matching of signal signature uses an improved Nearest Neighbors in Signal Space (NNSS) pattern recognition [15, 33]. In general, k -Nearest Neighbors (k NN) networks tessellate the input space by setting weights and thresholds of first layer hidden neurons. This is done by calculating the boundaries of space containing points nearest the training pattern. Subsequent processing layer handles the classification for the unknown input pattern [84]. A BP neural network handles classification of signal signatures for the eventual assignment of location estimates in further processes. Three hidden layers of processing are applied. The first layer is an input layer that distributes all input patterns. Second layer provides feedback control input back to the first layer. The final layer forms variable threshold neurons for output classification. For a multilayer network, the output of one layer becomes the input of the following layer as shown in Figure 4.15. The equation that describes these operations is

$$\mathbf{a}^{m+1} = f^{m+1}(\mathbf{W}^{m+1} \mathbf{a}^m + \mathbf{b}^{m+1}), \text{ for } m = 0, 1, \dots, \mathbf{M}-1, \quad (38)$$

where $\mathbf{M}$ is the number of layers in the network. The neurons in the first layer receive external inputs, such that $\mathbf{a}^0 = \mathbf{p}$, which is the starting point. The outputs of the neurons in the last layer are considered as the network outputs, such that $\mathbf{a} = \mathbf{a}^3$ in a 3-layer neural network. In the BP algorithm, a set of proper network behavior is presented as:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\}, \quad (39)$$

where $\mathbf{p}_i$ is an input to the network, and $\mathbf{t}_i$ is the corresponding target output. The network output is compared to the target as each input is applied to the network. The algorithm then adjusts the parameters to minimize mean square error. A vector composed of the elements of the i th row of $\mathbf{W}$ was defined as:

$$\mathbf{w}_i = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix} \quad (40)$$

as well as the vector

$$\mathbf{V} = \begin{bmatrix} \mathbf{w}_1 \\ b \end{bmatrix} \quad (41)$$

Thus the network mean square error is

$$F(\mathbf{x}) = E[\mathbf{e}^T \mathbf{e}] = E[(\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})] \quad (42)$$

The mean square error can be estimated by

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k), \quad (43)$$

where the expectation of the squared error is being replaced by the square error at iteration k . The net input to layer m is an explicit function of the weights and bias in that layer:

$$n_i = \sum_{j=1}^{S^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m. \quad (44)$$

Therefore,

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \quad \frac{\partial n_i^m}{\partial b_i^m} = 1. \quad (45)$$

If the sensitivity of $\hat{F}$ to changes in the i th element of the net input at layer m is defined as

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m}, \quad (46)$$

then

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} = s_i^m a_j^{m-1}, \quad \frac{\partial \hat{F}}{\partial b_i^m} = s_i^m. \quad (47)$$

By applying the steepest descent algorithm for the approximate mean square error, i.e.

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m}, \quad (48)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m}, \quad (49)$$

where α is the learning rate, then the following results are obtained:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1}, \quad (50)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m. \quad (51)$$

In matrix form this becomes:

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T, \quad (52)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m, \quad (53)$$

where

$$\mathbf{s}^m \equiv \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix}. \quad (54)$$

To compute $\mathbf{s}^m$, chain rule is applied. BP is used because the process of computing the sensitivities is a recurrence relationship in which the sensitivity at layer m is computed from the sensitivity at layer $(m+1)$. For BP, the following applies:

$$\mathbf{s}^M = -2 \dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}), \quad \text{where } \mathbf{M} \text{ is the last layer} \quad (55)$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, \quad \text{where } m = M-1, \dots, 2, 1 \quad (56)$$

and

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{s^m}^m) \end{bmatrix} \quad (57)$$

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m}. \quad (58)$$

An adapted k NN algorithm calculates the degree of match between calibrated RSS dataset and online RSS signature. Euclidean distances measure the similarity between two points $M = (m_1, m_2, \dots, m_n)$ and $L = (l_1, l_2, \dots, l_n)$:

$$\begin{aligned} d &= \sqrt{(m_1 - l_1)^2 + (m_2 - l_2)^2 + \dots + (m_n - l_n)^2} \\ &= \sqrt{\sum_{i=1}^n (m_i - l_i)^2}. \end{aligned} \quad (59)$$

The calibration data are compared with the signal strength measurements in the mobile device during the adaptive localization phase using:

$$d_j = \sqrt{\sum_{i=1}^n (c_j^{AP_i} - s^{AP_i})^2}, \quad (60)$$

where $c_j^{AP_i}$ represents RSS values of AP_i at point j in the signal strength radio map,
 s^{AP_i} is the measured RSS value of AP_i ,
 N refers to the total number of APs.

Small d_j distance value between a target location and an identified location implies close proximity between two location comparisons. The pattern matching algorithm implementation is as follows:

```
int PatternRec::LeastDistance(int ReNum){
    double min = 10000;
    for(int d=0; d<10; d++){
        if(MatchData(type[d].Feature, InputNum)<min){
            min = MatchData(type[d].Feature, InputNum);
            ReNum = d;
        }
    }
    LocNumber = ReNum;
    return ReNum;
}
```

Conventional NNSS method include RSS values from every known APs without considering RSS variations from each AP. However, studies [13] had shown that individual variations in RSS of a single AP can substantially influence d thereby affecting the final quality of classification. Thus, k lowest d values are often considered in estimating true deviation. The choice of k depends on the size of the dataset. A large k reduces the effect of noise on classification but decrease the ability to distinguish boundaries between classes. An enhancement to NNSS [85] algorithm is integrated to take into account the threshold for RSS variation around its mean, θ and the maximum number of APs from which RSS varies beyond threshold, τ :

```
Input: sorted neighbour list of size  $S$ , where  $s \geq k$ 
begin
for  $m = 1$  to  $k$ 
begin
    count=0
    for  $i = 1$  to  $n$ 
    begin
        if  $(|c_j^{AP_i} - s^{AP_i}| > \theta)$ 
            count = count + 1
    endfor
    if (count  $\geq \tau$ )
```

```

        remove neighbor m from neighbor list
    endfor
    if (less than k entries in neighbour list)
        return old list
    else
        return updated list
    end

```

The enhanced NNSS technique is applied to the leading k entries in the input neighbour list sorted according to Euclidean distance. Using cross-validation techniques, $k = 3$ was determined to be an optimal distinguishing parameter for boundary determination purposes in a typical WLAN setup of 4-8 APs detected throughout all location grids.

Location estimation can be achieved by looking at probability of a random variable observation at individual site. Studies [13] had shown that RSS readings follow an approximated Gaussian distribution model represented by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (61)$$

where μ is the mean and σ^2 is the variance. Using central limit theorem, a large number of random signal observations from nearby APs can be approximated to a normal distribution making it ideal for localization. During the adaptive localization phase, the mathematical problem is to establish the probability of the device being at the stated location. Using Bayes' Theorem, this probability can be calculated by:

$$p(l_r | X) = \frac{p(X | l_r)p(l_r)}{p(X)} = \frac{p(X | l_r)p(l_r)}{\sum_{l'_r \in L} p(X | l'_r)p(l'_r)}, \quad (62)$$

where $p(X | l_r)$ represents likelihood function of obtaining observation X at location l_r ,
 $p(l_r)$ represents prior probability of location l_r ,
 $p(X)$ represents normalizing constant.

The approach here is to utilize Maximum A Posteriori (MAP) estimate in bayesian probability model to assign a probability for each possible location given an observation. This provides a graphical representation of a joint probability distribution that declares dependency relationships between random variables. l follows a normal distribution, $p(l_r | X)$ depends only on the likelihood $p(X | l_r)$. Therefore, l can be chosen to be the estimation result when its likelihood $p(X | l_r)$ is at its highest. For better estimation, weights are assigned to reference locations derived from their normalized likelihood.

$$\hat{l} = \sum_{r=1}^R l_r \cdot \hat{P}_r = \sum_{r=1}^R l_r \cdot \frac{P(X | l_r)}{\sum_{r=1}^R P(X | l_r)}, \quad (63)$$

where $\hat{l}$ is the location estimate,
 l_r refers to the r -th reference location,
 R is the total number of reference locations,
 $\hat{P}_r$ represents a normalized likelihood with respect to all posterior probabilities.

Parameters of the Gaussian distribution such as the mean vector and covariance matrix, are calculated by MAP and stored for each l_r during the collection phase using:

$$\hat{\theta}_{MAP}(X) = \arg_{\theta} \max \frac{p(X | l_r) p(l_r)}{\sum_{l'_r \in L} p(X | l'_r) p(l'_r)}. \quad (64)$$

MAP estimate is a regularization of maximum likelihood estimation by employing augmented optimization objective to incorporate a prior distribution over the set of possibilities. The MAP localization estimate works as follows:

- 1) Measure distance from reference node to receiver.
- 2) Determine sum of squares differences in measurements and the true Euclidean distances of all the reference nodes for each grid point location in the localization space.
- 3) Choose the grid point location with the least value of the sum as the location of the receiver.

A grid resolution 100 times higher than the dimensions of the localization space is used to determine the location of the receiver. Changing environmental conditions due to signal drag movement effects leads to changes in received signal strength. A small scale compensator detects small scale variation if the distance between estimated location and previous user location is within a stated threshold. The threshold parameter is determined by the user's speed and location update. Whenever small scale variation is detected, it compensates the received signal strength vector entries. The following equation illustrates the action of the compensator on the signal strength vectors:

$$X_1 (1+y), X_2 (1+y), \dots, X_n (1+y), \quad (65)$$

where $y \in \{-d, 0, d\}$ and d is the proportionate change applied to the signal strength. By changing the component link to the strongest reference node, the compensator can obtain

sufficient results with low computation. After which the module will re-estimate the location.

4.3 Cluster based vs Filter based

Countering signal drag requires the differentiation of outdated or unreachable reference nodes from active nodes. Within a WLAN environment, filter based techniques required less computation than cluster based solutions as they only requires iterative removal of the least important results until a desired outcome is achieved. Previous observations had indicated that moving receiver in a WLAN is more affected by access points positioned at the forward or backward direction of the movement path than access points placed at the side. Thus, receiver moving towards/away from the general heading of known access points should factor in the influence of these access points during such movement. The adaptive cluster selection mechanism helps to match radio signature in devices experiencing signal drag due to movement. It does so by minimizing data search complexity by reducing search space required to return a result and offers variable matching in different movement conditions. Such flexibility is not available in filter based solution as they treat all movement conditions equally. The module categorizes locations into a cluster that shared the same set of APs. During adaptive localization, the module only searches the relevant cluster containing the detected APs instead of probing the entire radio map search space for a match. Each location cluster has at least q APs detected. This minimum set of q relevant APs is required to provide sufficient information for location determination as not all APs provide the same signal profile contribution to a single location. The parameter q is dynamic. By varying q , a series of related location cluster is formed which are adaptive to changes in signal similarities in different regions. Table 4.1 shows the effect of varying q on the coverage that the clusters has on the search region.

Table 4.1 Effect of q on cluster coverage for search region

	q=1	q=2	q=3	q=4	q=5
No. of clusters formed	100	12	9	3	0
Largest cluster coverage	N.A.	20%	13%	14%	N.A.
Smallest cluster coverage	N.A.	6%	6%	9%	N.A.
Average coverage	N.A.	11%	8.40%	11.30%	N.A.

In this profile, $q=3$ was found to be most effective at reducing search boundary but by itself is not conclusive evidence on computational performance. Manual code instrumentation is applied to establish the runtime for each execution to evaluate the effect of the number of clusters and its size. The algorithm was adjusted such that only the tested cluster is executed repeatedly to determine the performance for various types of cluster.

Figure 4.15 Comparison of various q sizes ($q=2,3,4$ from left to right) used in the radio map cluster on computation performance. The vertical axis represents the computation time in milliseconds (ms). Horizontal axis represents the amount of executed runs.

Table 4.2 shows the average time taken for different type of clusters to complete the computation. Against the possible q cluster sizes, $q=3$ offers the best performance across different type of clusters except when cluster size is smallest.

Table 4.2 Average time taken to complete cluster computation (ms)

	First Cluster	Last Cluster	Smallest Cluster	Largest Cluster
$q=2$	17.5	19.9	18.6	16.9
$q=3$	16.6	19.3	22.7	16.9
$q=4$	20.5	21.2	20.5	19.4

Typical signal drag causes an average signal strength difference of -20dbm across different movement speed. This translates to a corresponding expansion/contraction in cluster coverage of approximately 10% in order to compensate for the difference. In this case, varying q to the optimal size can achieve the most optimal performance and experiments show an average of 7% decrease in computation time with the use of clustering. The performance gain could be even more substantial in congested Wi-Fi environment as it likely to enjoy greater benefits from clustering.

Figure 4.16 Average estimates error in signal matching localization performance for filter and cluster based methods.

Figure 4.17 Cumulative Error Distribution Functions for Cluster, WCL, RSS and GT based methods. Average distance error decreased statistically with the use of Cluster based solution.

Fig. 4.16 shows a comparison of the average location estimate errors in both filter based and cluster based methods. Adaptive cluster solution achieved a 57.9% average reduction in signal drag experienced by the receiver. Compared to filtered based solutions, there is a 23.1%, 65.6%, 29.3% improvement against WCL, RSS and GT based filtering respectively. It was observed from Fig. 4.17 that cluster based solution has better matching of location to signal strength distribution as movement conditions are mapped more accurately through this adaptive selection. Filter based solution performed poorer as the least likely results are always discarded even though it has some contribution to the

estimation. The use of adaptive cluster selection for radio map is ideal for reducing the detrimental effects arising from multipath and signal drag in proximity based application distribution.

4.4 Conclusion

This chapter described the challenges associated with mobile environment and the proposals to mitigate such problems. Most importantly, we identify signal drag as a key issue to be resolved for the implementation of a proximity based mobile distribution service. A marked improvement in reducing signal drag in a wireless medium is observed using cluster based solution instead of conventional filters. The proposed cluster based solution is more efficient in identifying the relative contributions from wireless data and this solution is incorporated into the localization aspect of the proposed proximity distribution system described in the next chapter.

CHAPTER 5

Proximity Distribution for Mobile Applications

Bearing in mind the unique situation of mobile computing, mobile applications requirements are significantly different from that of fixed distributed applications. The issue of distribution has been a major concern for location-based service developers. User downloads application onto their mobile phone in order to utilize their services. As more applications are developed, a significant number of applications become obsolete as they are rapidly replaced by newer content. User often only requires the application at the moment of need and often do not need to keep the application for the long term. This chapter would be dealing with the distribution of mobile services. The objective is to develop a novel application distribution manager which responds to location changes. Application distribution is reinvented by pushing related application to users when they are in a relevant service area. Applications are automatically removed when they are no longer needed once the mobile is outside the service area. The intention is to create a mobile service agent to distribute relevant service applications when in proximity to a smart space.

Current location aware applications apply their own location APIs to locate itself and make use of this location information to search for other information they are interested in. For example, Google implemented Google Map APIs on mobile phones to estimate a user's position and allows him to search for information like location of restaurants through its internet search service. Similarly, applications like rednano.sg also provide their services through this manner. Every service would require its own independent application to be installed in the mobile device. Research had shown that most users only use a small percentage of the applications they download on a regular basis. According to Pinch Media [86], of the 30 million downloads from Apple's App Store analyzed, just 30 percent of people who bought an application actually use it the day after it was purchased. The numbers plunge further with increasing length of time after download.

5.1 iSARIS Distribution Platform

This section describes a prototype platform to tag applications to smart service areas for location aware applications. Intelligent Service Area and Region Identification System (iSARIS) uses cell tower identification and wireless radio maps to identify smart physical space regardless of the actual shape. When the user's mobile phone enters a service area, it can download relevant applications simultaneously and remove these applications when the user leaves the service area. The objective of creating iSARIS is to integrate sensing technologies onto a single layer where the strengths of each can be maximized. Localization services are migrated to a single platform where mobile entity switches smartly to the correct technologies for the relevant environment. Existing service area identification system is tweaked using adaptive cluster selection to improve performance. This solution enables a mobile device to passively observe surrounding radio signals to recognize Service Region (SR) and Service Area (SA) smart space. Applications "reside" in such a smart space and are ready to be installed to any interested mobile device that enters a SA.

In mobile telecommunication network, BTS towers are situated where three hexagonal coverage cells converge in a cellular map. Each tower has three sets of directional antennas aimed in different directions. Large cells are subdivided into smaller cells for higher volume areas. A SR consists of mobile coverage cells servicing the vicinity of the building. A background service on the mobile device observes telecommunication exchanges between the device and its associated BTS. Power used to monitor signals is kept minimal as the process runs parallel to normal telecommunication operation.

Figure 5.1 Mobile enters a SR from public area as it nears complex (Left). Mobile accessing SA defined under Wi-Fi coverage (Right).

A coarse location estimate obtained through the background service activates the discovery service which flagged an alert whenever an operating SR is identified. This triggers a secondary positioning system suitable for indoor environment. Many buildings today are also equipped with WLAN for internet access. APs placed at strategic locations create Wi-Fi hotspots for users to access via Wi-Fi enabled devices. SA are created using radio mapping, which marked a clear definition of smart space, where a particular service is valid. A single SR may contain multiple SAs, which can also overlap with other SAs. SA allows individual service space to be built for providers to offer individualized services on a smaller physical scale. Once inside a SA, a mobile service agent determines the relevant service application to provide for users. Mobile automatically downloads and installs the applications in preparation for use by the user. Upon leaving the SA, applications can be removed by the service agent to free up resources for future use.

5.2 Implementation on Windows Mobile Devices

A shopping mall service system is illustrated here as an example on how the iSARIS platform can be used for mall customers interact with smart services provided by the mall operator and shop vendors. The system implementation defined the shopping complex and its immediate surroundings as a SR. Upon entering the SR, customer receives an automated alert from the complex service server listing the offerings available. While roaming within the SR, the mobile device can access mall services, such as directory information, promotion events, mall promotions and helpdesk services. This is delivered through the shopping mall WLAN infrastructure. Once within the SR, the mobile device also uses the Wi-Fi module to improve positioning accuracy and identifies any available SAs. Vendors create SAs to define smaller physical spaces to offer their services. The mall Wi-Fi infrastructure handles vendor content and decides whether to push service to customer when they enter a SA. Services available in SA can include discount notifications, pricelists, shop advertisements and vendor specific promotions. Mobile devices connect to the position and data server via the wireless APs installed in the shopping mall.

Figure 5.2 Typical deployment and interaction between entities

5.2.1 Component Systems

The shopping mall system is divided into 3 major component systems. Each of these subsystems contains individualized modules that handle specific tasks. Figure 5.3 shows a schematic of how the different components interact.

Figure 5.3 Shopping mall service using the iSARIS Platform

A switching module is responsible for toggling between primary to secondary localization services during the detection of a SR. The discovery background service is handled by this module. The process of switching between positioning technology is automated and with little intervention from user. Mobile automatically diverts location recognition from telecommunication cell identification to a Wi-Fi based pattern matching when the mobile device approaches the shopping mall. The interface module contains browsing and selection services. Such functions include accessing information from mall and vendors,

interactive service selection and service registration. A mobile client user interface is implemented to deliver these functions to the user. The operation module is an html based system to allow vendors to control the offerings available to their customers. The module accesses the shopping mall server through an internet based login system. Vendor control information to the customer using simple add, delete and update tools. It also allows vendors to define SA boundary through the shopping mall location service in accordance to its needs. The location module handles mobile client location related events. It monitors the creation and dismantling of vendor designated SAs and alerts mobile that enters into the physical locality of a defined SA. Communication module is responsible for the information transmission between shopping mall, vendor services and the mobile client. Direct link between the mall and the vendors are established using html based client-server link. The mall acts as the service agent to link mobile client and vendors using its WLAN infrastructure. It can also connect directly with mobile client itself to provide directory and mall specific information. The management module is responsible for handling user events within the system. This module is situated at the mall server end and manages basic vendor information (eg. vendor identification, sale items, and service range), discounts information, promotional materials and helpdesk information services.

5.2.2 Position & Data Server Design

Position and data handling applications are programs hosted on the mall server. The server handles tasks like receiving signal signature from mobile devices, running the location algorithm to establish mobile device position and determining the relevant vendor web data to send to customer. Administrators of the mall server manage the position and data services. The position server receives mobile devices' signal strength parameters via Transmission Control Protocol (TCP) connections. A location algorithm is used to calculate an estimated position fix for each client. The estimated position will be used to determine whether a device is in the region of a vendor-defined SA before the data server send the corresponding vendor website to the client's mobile device across a User Datagram Protocol (UDP) connection as seen in Figure 5.4.

Figure 5.4 Allocation of SA on position server

The implementation code contains two main parts. A static `ScanThreadFunc()` function is used to create a scanning thread responsible for wireless access points scanning in the shopping mall. Scanning for Wi-Fi access points is handled as a looping task in this thread. The thread is created when the shopping mall SR is identified and terminated whenever the Wi-Fi connection is disabled. The acquisition of access points' signal strength is handled in the `WifiScanConnect` class by a `ScanWifi()` member function.

5.3 Implementation on Android Devices

Android is a software stack by Google and the members of the Open Handset Alliance [87]. It consists of a modified Linux kernel and C/C++ libraries available through an application framework. It provides management of runtime and applications services for resource constrained devices. Its components portability and open source concept suits the need for embedded systems. Such features are highly desirable for manufacturers, service providers and developers as it provides a large spectrum of application possibilities [88-90].

Figure 5.5 Android Architecture

The main vehicle for application development in Android is the Java programming. Google Android runs a Linux kernel to manage the core system services and provides the abstraction layer between the underlying hardware and the rest of the software stack. It manages device drivers, resources, power, security and networking interfaces with the underlying hardware. C/C++ libraries such as SQLite, Graphics libraries OpenGL ES, media framework and webkit layout engine are made available to developers via the application framework. The Android Runtime includes the core libraries which provide most J2SE libraries in addition to Android's own Java libraries and the Dalvik Virtual Machine (VM). Applications written in Java source codes are first compiled into Java bytecode and subsequently into Dalvik bytecode (.dex format) for execution on the VM. Dalvik VM is optimized for mobile operations and provides Just-In-Time compilation for software runtime in the device. Multiple instances of VMs can be created for each Java based application to access the core libraries without significant detrimental influence on efficiency.

Android application framework enables reuse and replacement of software components. It manages the necessary classes to develop applications and provides the relevant abstraction for hardware access through an API. The Android Java API's main library includes telephony, activity managers, content providers, resources, locations and user interfaces. All Android applications, both native and third party, are built on the application layer using these APIs. Applications can access locally accessible private database through SQLite, which is a relational lightweight file based database.

5.3.1 Basic Android Operations

An Android application is defined by the contents of the `AndroidManifest.xml` file. All activities, communication layers, entry points, intents and permissions are declared through this file. Four basic foundations [88] are needed to build up an Android application:

- 1) Activity: a subtask that application can do
- 2) Service: a background process that does not require a user interface
- 3) Intent receiver: a reactive object that is launched to handle an assigned task
- 4) Content provider: a basic superclass framework that handles/stores data

A central feature of Android application is the reuse of previously developed components of other applications without the need to add in or link to the external source code. Developers decide how to publish and share activities or determine how to handle services and data between applications that has security restrictions. This is possible as Android applications do not have a single entry point for execution. Android applications are modeled as a package consisting of a collection of components that share a set of resources. This includes the compiled Java code, databases and resource files. It is bundled using an `aapt` tool into an archive file similar to a zip file and marked by an `.apk` suffix. It is then used in the distribution of software applications on Android based mobile devices. The main idea is for the package to contain the essential elements that the system can choose to run as and when needed. This enables developers to incorporate out-of-the-box components to their creation or add new functionalities to them.

5.3.2 Function Overview

Location of the user is used to employ a distribution mechanism for the mobile services. A selection menu listing suitable applications is provided so that user can manage the applications according to preference. A list of services relevant to the smart space is pushed to the user interface via a recommendation database linked to applications distribution channels such as the Google Play (previously Android Market). This recommender identifies current smart space characteristics and associates relevant applications in order of context and user reviews. Users can access and use this interface as if the applications were installed locally.

The desired applications selected from the generated recommendation list are downloaded for use when device is within service area. User device location is monitored to determine

if the user stays within the confines of the smart space and continue to require the service of the application. When the user leaves the smart space, he/she could manually remove the application or leaves it to the system to handle the removal. System removal allows the management of service provisioning processes in an efficient manner and avoids memory constraint problems.

Figure 5.6 Class diagram illustrating the dependencies between system classes

Figure 5.6 shows the dependencies between classes in the application distribution system. UserInterface and MobileDevice are boundary classes that model interactions between the system and its actors. Such interactions include information and requests from users and external systems. Application, AndroidMarket, Location and HomeScreen entity classes model information with long and persistent lifespan. These classes model information and associated behavior of phenomenon or concept such as an individual, object or event. LocationListener, Connection and DBManager are control classes representing coordination, transactions, sequencing and control of objects. In addition, they encapsulate control related to particular use cases.

5.3.3 Activity Class

Each application runs in its own Linux process. Whenever a code is executed, the OS determines how and when to run the processes based on current system resource state. Each process is allocated its own VM such that an application can run its programming code in isolation from one another but there are mechanisms in place for different processes to interact with each other. Each user interface screen is represented by an Activity class with its own life cycle. It uses an Intent object to move between states.

Intent can activate activities, services, and broadcast receivers using predetermined parameters. To receive requests from other components, filters are registered at the activities framework. When an intent object is launched, the framework will match the relevant components and let users choose which to run. Activities in the system are managed as an activity stack. Whenever a new activity is initiated by a user, that activity is positioned on the top of a stack. Previous activity always remains below the new activity in the stack. It does not come to the foreground until the new activity had exited. The activity manager controls this life cycle. A task is an activity stack which contains several activities that share the same affinity. An activity has the basic four states:

1. If the activity is at top of the stack, i.e. in the foreground of the screen, it is deemed to be active or the current running activity.
2. Whenever an activity does not have the focus of the user, it is paused. A paused activity is still alive but can be killed by the system in situations of low memory. For example, a non expanded activity in the background but nevertheless still visible to user.
3. An activity is stopped if it is completely covered by another activity. The activity retains member and state information but is no longer noticeable to the user. For example in cases where the window is hidden. It is often killed by the operating system whenever memory is needed somewhere else.
4. The system can drop any activity from memory if an activity is paused or stopped. This is done by giving a finish command or by killing the processes. When required again by the user, the activity must be restarted and restored to its previous state.

Figure 5.7 shows an Activity state paths. Rectangular blocks represent the callback methods which is implemented to carry out operations when an Activity moves through different states. Ovals blocks are the different type of states an Activity can be in.

Figure 5.7 Flow diagram illustrating the state paths of Activity classes

5.3.4 Service and System Classes

A Service is a component object in Android representing a lengthy runtime operation. Each service class has an equivalent `<service>` declaration in the package manifest and can be started using `Context.startService()` or `Context.bindService()`. Services run in the main thread of the hosting process. An operation should create its own thread to do any work that is CPU intensive or require the service to do any blocking operations. The `IntentService` class is part of the standard implementation of `Service`. It requires its own thread to schedule any work to be done. When a `Context.startService()` call is made, the OS retrieves the service and then call the `onStartCommand(Intent, int, int)` method with the arguments given by the client. Multiple calls to `Context.startService()` do not nest. Clients use `Context.bindService()` to connect a persistent connection to a service.

5.3.5 Smart Space Fundamentals

Physical locations are managed smart spaces defined by a location database. An example would be the physical boundary of a shop inside a mall. Applications suitable to the user for the current context are pushed to mobile device when the user is within a service area and removed automatically when the user leaves.

The Android application framework provides for an `android.location` package which supplies the relevant APIs to access best known location information currently active and available on the operating system [89]. Important classes present in this package include:

- 1) *LocationManager*: provides location service and the ability to obtain the best available Location Provider according to a set of criteria. Proximity alerts can also be scheduled by the manager.
- 2) *LocationProvider*: abstract superclass for location providers. Periodic updates are sent to update and report the latest geographical location.
- 3) *LocationListener*: provides the callback methods to handle location change. The listener object requires prior registration with the location manager.
- 4) *Criteria*: provides application the means to select suitable location provider and gain access to its properties. These properties include power consumption, ability to report altitude, accuracy, speed, direction and costs.

Applications use these services to obtain periodic updates of device geographical location. In addition, it can also initiate application-specified intent whenever the device enters the proximity of a targeted location. This class is retrieved through `Context.getSystemService()`. The location manager object can be obtained with `Context.getSystemService` method using the `Context.LOCATION_SERVICE` parameter. Device current position is collected using the location API:

```
LocationManager lm = (LocationManager)
getSystemService (Context.LOCATION_SERVICE);
```

Location Based Service (LBS) providers within a mobile device can be retrieved using `getProviders()` method in the Location Manager Class. The APIs available for mobile devices include Cell Tower Triangulation, Wi-Fi Mac addresses and GPS/aGPS sources. The location identification process is carried out through the following steps:-

1. The latest location parameters are retrieved as given by the providers.
2. Acceptable results are recorded.
3. A more precise location based on update timing and accuracy is measured among recent ones and the best last known position.

GPS/aGPS accuracy is known to be up to several metres depending on received GPS signal quality and connectivity to assisting tower. WiFi has a similar accuracy to the access point range of a typical WiFi router which is around 20m. Cell ID accuracy depends on the cell tower density and also the availability of data in Google's cell tower location database. Coverage for WiFi and cell ID location databases varies by location. Coverage and accuracy can be improved over time by increased use of location-based services. Location parameters received has up to 15 digits precision but only 8 decimal places are used in this implementation to store smart space latitude and longitude as it already give an accuracy equal to 1 mm in distance which is more than sufficient for the required purpose.

LocIdenService is the main control class that records current Location parameters. It compares the current location with the parameters stored in the database using the Haversine Formula to decide whether the current location is within the radius as stated in the databases. If found to be in the coverage, a notification about the current Smart Space is made to the user interface.

Haversine Formula

Difference in longitude = longitude2 - longitude1

Difference latitude = latitude2 - latitude1

$$a = (\sin(\text{difference in latitude}/2))^2 + \cos(\text{latitude1}) * \cos(\text{latitude2}) * (\sin(\text{difference in longitude}/2))^2$$

Developer can set its own criteria for accuracy and resource management which is done by varying the frequency and distance between updates using `getBestProvider()` method. The `LocationProvider` class provides features about the LBS providers such as whether it supports altitude, bearing, speed information and the costs of obtaining such information. Using the `getLastKnownLocation()` method, the location manager provides a baseline location which may not be current but can be a useful start point for further updates. A service discovery service implemented by the `LocationListener` class retrieves the current

tower and neighboring cell parameters from mobile phone. Depending on the network available, location can be obtained from the `GsmCellLocation` or `CdmaCellLocation` classes. The `onCreate()` event handler create an instance of the `TelephonyManager` class which in turn obtain an instance of the `GsmCellLocation` class:

```
public void onCreate (Bundle savedInstanceState) {

    super.onCreate (savedInstanceState);
    setContentView (R.layout.main);

    TelephonyManager tm =
        (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
    location = (GsmCellLocation) tm.getCellLocation();
}
```

To obtain the device's cell ID, LAC and PSC (For UMTS network), the `getCid()`, `getLac()` and `getPsc()` methods are used. With these data, a separate process resolves this information into latitude and longitude. In order to obtain latitude and longitude information from the cell ID and the LAC, a backend service access open cell ID database to generate the coordinates. Reverse geocoding service is used to translate coordinates to actual addresses. This defines the major service region in which the mobile has entered.

Figure 5.8 Location data are sent to google map servers. Data collected using crowding sourcing techniques are used to build geographic zones based on common unique cell IDs.

A `SSProcessor` control class triggers activities based on the Location identification results of the `LocIdenService` class. To identify individual smart spaces, the background service queries a pre-constructed SQLite database stored locally on the device. If the user is found to be in the Smart Space, it calls the applications querying from another database, and when the user is away from the current Smart Space, it calls the destroy method for the

service which triggers uninstallation processes. By utilizing crowd sourcing techniques, geographical zones are established where applications can be downloaded to provide the related context services for the zone.

5.3.6 Database classes

Localization sensitivity is managed by the SQLite database linked to google map to ensure efficient determination of suitable sizes for smart space boundaries. Applications are managed in a recommender database and grouped according to zones for classification. Recommender list push to user are based on location polling methods. Whenever the location realizes that a service area is of interest to the user, it triggers an intent for a recommender list that is customized for the area. Polling for service areas using location intents reduces the need for constant active location monitoring.

Figure 5.9 A list of suitable applications for the particular service area is pushed to the device on entry into smart space service areas. Applications are installed on and removed from mobile device as it transits between spaces.

The system requires two tables: one table to store location resources and another to store application resources. In android implementation, database is private to each application. As such, an activity or service cannot open a database in one thread to pass the handle off to another thread to use. To create a database, a SQLiteOpenHelper class is extended over which the onCreate() and onUpgrade() methods are used to import the external SQLite database created. Only one table can be used in a class that extends Activity per object.

A SQLiteOpenHelper helper class manages database creation and version management. A subclass is created using onCreate(), onUpgrade () and onOpen(). The subclass is in charge of opening database if a database exists and creating the database if it does not. It also upgrades the database as and when it is necessary. This implementation makes it easy for ContentProvider to defer opening and upgrading operations on the database until first use so as to avoid blocking the application startup with lengthy database upgrades.

Table 5.1 shows the lists of suitable applications that are suitable for smart spaces within a larger physical region. The recommender takes user preferences into account before generating the list. Tagging applications to smart spaces creates a sense of context awareness for the user without the conscious effort involve in searching for relevant service applications.

Table 5.1 Distributable Android application list by location zone.

<i>Area</i>	<i>Zones</i>	<i>Applications</i>	<i>Examples</i>
Mall	Stores	Fashion, Electronics	ShopSavvy, London Fashion Trend
	Restaurants	Food reviews	buUuK
	Customer Service	Directory	ShopNow, SG Malls
	Movie	Ticketing	Flixster
Airport	Departure & Arrival halls	Flight status	iChangi, FlightStats
Museum	Restaurants	Food reviews	buUuK
	Exhibits	Information	Wikidroid
Tourist site	Attractions	Walking tours, Places of interest	Guidepal Singapore city guide, Lonely Planet's Compass guide
		Public transport	Taipei Bus Map, SG Buses
	Hotels	Bookings	Priceline Hotel Negotiator

A SQLiteDatabase exposes methods to create, delete, execute SQL commands and carry out related database management tasks. Database names need to be unique in an application, although they do not need to be unique across different applications. In this implementation, there are two databases with the following schemas:

Location.db

```
CREATE TABLE [Location] (
```

```
[_id] integer PRIMARY KEY AUTOINCREMENT,
[loc_name] text,
[radius] REAL,
[latitude] REAL,
[longitude] REAL);
CREATE TABLE android_metadata (locale TEXT);
```

Application.db

```
CREATE TABLE [Application] (
[_id] integer PRIMARY KEY AUTOINCREMENT,
[loc_name] text,
[app_name] text,
[app_url] text,
[apk_package] TEXT,
[app_status] TEXT,
[app_descrp] TEXT);
CREATE TABLE android_metadata (locale TEXT);
```

To push a database externally into a class, an Android_metadata table is needed. The databases are flooded with the values recorded for the location resources and the applications relevant to the stated location. The created databases will be added to the assets folder of the application and then used by DBQueryHelper Class to extract data or update data in the tables. 2 extension classes are created from the SQLiteOpenHelper class: DBLocation and DBApplication. The DBQueryHelper is a standalone class that holds the objects of DBLocation and DBApplication. It carries methods for queries to assists the two modules in retrieving the location parameters to check the users' vicinity in the smart space. The queries will return the list of applications relevant to the smart space and the flag for uninstallation if needed.

5.3.7 Other classes

The User Interface module is the boundary class for user interactions. Its purpose is to notify user of the current smart space, receive the user selection and install the relevant applications. Java elements like Button, CheckBox and TextBox are used to construct a proper interface. AppScreen Class extends Activity class. It receives location parameters from LocIdenService class to create the relevant notification on the user screen through a

UserInfo Class. UserInfo Class handles notifications to users about the current smart space it is in by giving a notification if the location parameters matches those found in the DBLocation database. On confirmation that the mobile device is indeed in a smart space, the application recommendation list is retrieved from the DBApplication database and listed in a TableLayout with CheckBox in anticipation for user application selection.

Figure 5.10 List of suitable applications for the particular service area is pushed to the device on entry into smart space service areas. Applications are installed on and removed from mobile device as it transits between spaces.

A DI_Install control class is implemented to handle installation processes whenever a user chooses an application to install. Using the apk link available in the DBApplication database, it creates a temporary file to house the application retrieved from the online server in the application cloud. A openFile() method starts the installation process for the downloaded file. File deletion option is also provided to clear the cache once the process is completed. After installation, the status of the application field in the DBApplication object is flagged. This would assist the subsequent uninstallation process and help to retrieve only applications that have not been previously installed when the user revisits the same smart space again. To uninstall, the system uses the SSProcessor to trigger the

Uninstallation process by calling the onDestroy() method. This will uninstall the application using the package name received from the DBQueryHelper Class.

5.4 iSARIS Smart Space Efficiency

The iSARIS system uses circle shapes to create smart spaces. These circles are used in location database to create the smart space Service Area (SA) for use to compare against sensors readings. SAs can take on any shape using circle as the basic element. By varying the size of the circles, the location coverage can be optimized as seen in Figure 5.11.

Figure 5.11 Overview of smart space creation at Takashimaya, Orchard. Accuracy of smart space representation improves when more circles are used as they approximate the location shape better.

Figure 5.12 Average application recognition rate in each smart space created by varying number of circles for a few locations in Singapore

Figure 5.12 represents the application success rate when location smart space is composed of different number of circles. Onsite testing was carried out at various locations in Singapore to gather data on the efficiency and optimal size of smart space circles. The graph plots the application success rate based on the number of circles used to demarcate the location. The amount of false positive decreases significantly when more circles are used. This is because of lesser area which does not encompass the building and cause a decrease in the amount of computation needed. The application took an average of 4 attempts to determine the accurate smart space during testing when a single circle is used to create the smart space. Increasing the number of circles helps to improve the application success rate. A reason behind this increase in efficiency is because of the increase in the number of locatable smart space centers coordinates in the proximity. When creating smart spaces, it is essential to develop a good number of overlapping circles mapping for each location. A large number of overlapping circles actually increases the chance of the mobile location sensors picking up a coordinate that falls into one of the component circles of the smart space.

Figure 5.13 Different radii size affects location recognition. In general, smaller circle used in database for SA is more ideal in matching mobile location sensor readings.

The radius of the circle also performs an important function of determining the exact location of the mobile. Figure 5.13 illustrates the results obtained when smart spaces of different radii are used in some SAs. Overall, depending on the size of the smart space in question, circle covering $1/7^{\text{th}}$ to $1/8^{\text{th}}$ of the entire smart space area provides the

maximum successful database enquiry match thereby improving the application success rate. This works out to be between 25m and 100m for most commercial buildings in Singapore.

5.5 Conclusion

This chapter explained the prototype proximity based software distribution platform that tags applications to smart service areas. It describes a detailed explanation of its implementation and benefits of such a system. The software prototype was tested in a real world setting using windows based and android based platforms and the results showed good recognition rate for areas of interest using user's context and interest.

CHAPTER 6

Conclusion and Future Works

Current software distribution model requires user to download specific applications in order to use a service. From consumer research, it is known that majority of applications downloaded to a mobile device are not used on a regular basis. To minimize inefficiency, the iSARIS platform implements a different application distribution model from conventional solutions. The smart space feature in iSARIS provides proximity based application distribution on mobile handsets by only sending software applications to user in the space when it is needed based on the context.

This thesis described a multi-layered context delivery architecture, which uses a context modelling technique that quantify QoC parameters. This context refinement uses further sub processes to enhance context object with the view of refining it for mobile computational needs. In the mobile space, the optimal localization techniques for different situations are determined by how radio wave transmissions are affected in different environments. Some of the commonly known techniques may work well in certain situations but fall short of requirement in others. It is important to look at the challenges associated with wireless localizations. This thesis had examined the different detrimental aspects of wireless medium and how these aspects affect localization performance. A novel approach to tackle problems, like signal drag and multipath effects using Adaptive Cluster Selection, to determine suitable radio maps used in recognition. A significant improvement in performance was observed over conventional filter based techniques.

Most mobile communication technologies do not account for localization services in its design but mobile localization is highly sought after. As a result, many such systems are built upon existing infrastructure even though they are not ideal. The ability to switch between localization systems is therefore beneficial for users. The iSARIS platform enables such benefits using a toggling mechanism that recognized when to switch to different localization services. Primary location service provided by cell identification

based techniques quickly identify a coarse region of physical space as a SR. This space can be a building, a group of buildings or other related entities, where cell identification level of positioning performance is no longer acceptable. A secondary positioning service is required to provide additional performance. iSARIS uses Wi-Fi radio mapping to show further smart space segmentation in the form of a SA, where independent service provider can make use of to provision their service offerings. By automating the transition process, the mobile offers better and more seamless user experience. This location based service provisioning design offers a simple and practical approach to reduce service engagement time and is suitable for mobile environment where quick and efficient means of switching localization control is required.

Applications development can be in a manner to fit the location context without dealing with the localization aspects of the problem. Less applications remained installed in mobile device as a result of using iSARIS platform and this indirectly reducing the interface clutter experienced by user. However, iSARIS has its own set of limitations. Download time is expected to be a significant setback to the overall user experience but it is likely to be mitigated in the future by the continuous improvement in data transfer speed as mobile broadband becomes more common. Future researchers can make improvements in identifying smart space by trying different neural networks to identify NNSS patterns. The current iteration uses simple back propagation that has some limitations. For example, the learning process requires input scaling or normalization which is not always possible given the dynamic nature of wireless data. This is also applicable to wireless system utilizing cloud computing concepts.

Author's Publications

Jon Tong Seng, Quah and Luo Ren, Lim, "Neural network based admission controller for proximity aware mobile services", *2009 International Joint Conference on Neural Networks IJCNN*, pp. 2580-2587, 2009.

Jon Tong Seng, Quah, Luo Ren, Lim, Hendri, Budi and Kim Teng, Lua, "Towards automated assessment of engineering assignments", *2009 International Joint Conference on Neural Networks IJCNN*, pp. 2588-2595, 2009.

Luo Ren, Lim and Jon Tong Seng, Quah, "Towards Context Information Refinement for Proximity Mobile Service using Quality of Context", *International Conference on Mobile Technology, Applications and Systems, Mobility Conference*, 2009.

Jon Tong Seng, Quah, and Luo Ren, Lim, "Proximity Mobile Services: Neural Network Based Connection Admission Controller and other issues", *IEEE International Conference on Networking, Sensing and Control ICNSC*, pp. 388-392, 2010.

Jon Tong Seng, Quah, and Luo Ren, Lim, "Nearest Neighbor Tracking for Proximity Aware Application", *2010 International Conference on Wireless Networks ICWN*, pp. 538-544, 2010.

Luo Ren, Lim, and Jon Tong Seng, Quah, "iSARIS: An Integrated Approach to Mobile Service Discovery and Tracking" *2010 International Congress on Computer Applications and Computational Science CACS*, 2010.

Jon Tong Seng Quah, and Luo Ren Lim, "Location based Application Distribution for Android Mobile Devices", *Wireless Communications WC2011*, pp. 25-31, 2011

Jon Tong Seng Quah, and Luo Ren Lim, "Location Cluster with Nearest Neighbors in Signal Space: An Implementation in Mobile Service Discovery and Tracking", *2011 International Conference on Wireless Networks, ICWN WorldComp2011*, pp. 25-31, 2011

References

- [1] Weiser M., Hot topics-ubiquitous computing, *Computer*, 1993; 26(10), pp. 71-72.
- [2] P. Bellavista, A. Kupper & S. Helal, Location-Based Services: Back to the Future, *Pervasive Computing, IEEE*, 7(2), 2008, pp. 85-89.
- [3] Cecilia M, Licia C and Wolfgang E, Mobile computing middleware. *Advanced lectures on networking*, Springer-Verlag New York, Inc., 2002, pp. 20-58.
- [4] A. Oulasvirta, Finding meaningful uses for context aware technologies: the humanistic research strategy, *Proceedings of the SIGCHI conference on Human factors in computing systems*, Vienna, Austria, 2004, 247-254.
- [5] Y. Cui & V. Roto, How people use the web on mobile devices, *Proceeding of the 17th international conference on World Wide Web*, Beijing, China, 2008, 905-914.
- [6] S. Hong, User Research and User Centered Design; Designing, Developing, and Commercializing Widget Service on Mobile Handset. *Human Centered Design, Lecture Notes in Computer Science*, 2009, 854-861.
- [7] B. Bergvall-Kåreborn, D. Howcroft & D. Chincholle, Outsourcing Creative Work: a Study of Mobile Application Development, *International Conference on Information Systems*, 2010, 1-15.
- [8] Barkhuus L, Brown B, Bell M, Sherwood S, Hall M and Chalmers M, From awareness to repartee: sharing location within social groups, presented at the *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, Florence, Italy, 2008, pp. 497-506
- [9] Bulusu N, Heidemann J and Estrin D, GPS-less low-cost outdoor localization for very small devices, *Personal Communications, IEEE*, 2000; 7(5), pp. 28-34.
- [10] Ali A, Latiff LA and Faisal N, GPS-free indoor location tracking in mobile ad hoc network (MANET) using RSSI, *RF and Microwave Conference, 2004. RFM 2004. Proceedings*, 2004, pp. 251-255.
- [11] Alessandro C, Matteo P, Carlo B, Fabio L and Massimo V, Proximity classification for mobile devices using wi-fi environment similarity, presented at the *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, San Francisco, California, USA, 2008, pp.
- [12] Jian Z and Durgin GD, Indoor/outdoor location of cellular handsets based on received signal strength, *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 2005, pp. 92-96 Vol. 1.
- [13] Wang Y, Jia X, Lee HK and Li GY, An indoors wireless positioning system based on wireless local area network infrastructure, *The 6th International Symposium on Satellite Navigation Technology Including Mobile Positioning & Location Services*, 2003.
- [14] L. Pei, R. Chen, J. Liu, H. Kuusniemi, T. Tenhunen, Y. Chen, Using Inquiry-based Bluetooth RSSI Probability Distributions for Indoor Positioning, *Journal of Global Positioning Systems*, 9(2), 2010, 122-130.
- [15] P. Bahl, V.N. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, 775-784.

- [16] R. Battiti, A. Villani, T. Le Nhat, Neural network models for intelligent networks: deriving the location from signal patterns, *The First Annual Symposium on Autonomous Intelligent Networks and Systems*, UCLA, 2002.
- [17] P. Castro, P. Chiu, T. Kremenek, R. Muntz, A Probabilistic Room Location Service for Wireless Networked Environments. *UbiComp 2001: Ubiquitous Computing*, 2001, 18-34.
- [18] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, J. Sievänen, A Probabilistic Approach to WLAN User Location Estimation, *International Journal of Wireless Information Networks*, 9(3), 2002, 155-164.
- [19] A.M. Ladd, K.E. Berkis, A. Rudys, L.E. Kavraki, D.S. Wallach, Robotics-based location sensing using wireless Ethernet, *Wireless Networks*, 11(1-2), 2005, 189-204.
- [20] J. Krumm, K. Hinckley, The NearMe Wireless Proximity Server, *UbiComp 2004: Ubiquitous Computing*, 2004, 283-300.
- [21] Krumm J and Redmond W, Probabilistic Inferencing for Location, *Workshop on Location-Aware Computing (UbiComp 2003)*, 2003, pp. 25-27.
- [22] Lawrence R and Rabiner A, A tutorial on hidden markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 1989; 77(2), pp. 257-286.
- [23] Roxin A, Gaber J, Wack M and Nait-Sidi-Moh A, Survey of Wireless Geolocation Techniques, *IEEE Globecom Workshops*, 2007, pp. 1-9.
- [24] Trevisani E and Vitaletti A, Cell-ID Location Technique, Limits and Benefits: An Experimental Study, presented at the *Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, 2004, pp. 51-60.
- [25] Deblauwe N and Ruppel P, Combining GPS and GSM Cell-ID positioning for Proactive Location-based Services, *Mobile and Ubiquitous Systems: Networking & Services*, 2007. *MobiQuitous 2007. Fourth Annual International Conference on*, 2007, pp. 1-7.
- [26] Kunczier H and Anegg H, Enhanced cell ID based terminal location for urban area location based applications, *Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE*, 2004, pp. 595-599.
- [27] Hofmann-Wellenhof B, Lichtenegger H and Collins J, *Global positioning System. Theory and Practice*. 1993.
- [28] Zaidi AS and Suddle MR, Global Navigation Satellite Systems: A Survey, *Advances in Space Technologies, 2006 International Conference on*, 2006, pp. 84-87.
- [29] McDonald K, The modernization of GPS: plans, new capabilities and the future relationship to Galileo, *Journal of Global Positioning Systems*, 2002; 1(1), pp. 1-17.
- [30] Richton B, Vannucci G and Wilkus S, Assisted GPS for Wireless Phone Location — Technology and Standards. 2002, pp. 129-155.
- [31] Sayed AH, Tarighat A and Khajehnouri N, Network-based wireless location: challenges faced in developing techniques for accurate wireless location information, *Signal Processing Magazine, IEEE*, 2005; 22(4), pp. 24-40.
- [32] Junglas IA and Watson RT, Location-based services, *Commun. ACM*, 2008; 51(3), pp. 65-69.
- [33] Bahl P, Padmanabhan VN and Balachandran A, Enhancements to the RADAR User Location and Tracking System, *Microsoft Research*, 2000, pp., DOI.
- [34] Bahl P, Padmanabhan VN and Balachandran A, A Software System for Locating Mobile Users: Design, Evaluation, and Lessons, *Microsoft Research Technical Report MSR-TR-2000-12*, 2000, pp., DOI.
- [35] Castro P, Chiu P, Kremenek T and Muntz R, A Probabilistic Room Location Service for Wireless Networked Environments. *UbiComp 2001: Ubiquitous Computing*, 2001, pp. 18-34.

- [36] Krumm J and Hinckley K, The NearMe Wireless Proximity Server. *UbiComp 2004: Ubiquitous Computing*, 2004, pp. 283-300.
- [37] Ranganathan A, Al-Muhtadi J, Chetan S, Campbell R, Mickunas D. MiddleWhere: A middleware for location awareness in ubiquitous computing applications. *Proceedings of 5th International Middleware Conference (Lecture Notes in Computer Science*, vol. 3231). Springer: Berlin, 2004; 397–416.
- [38] Massimo Ficco and Stefano Russo A hybrid positioning system for technology-independent location-aware computing SOFTWARE—PRACTICE AND EXPERIENCE *Softw. Pract. Exper.* 2009; 39:1095–1125
- [39] di Flora C, Ficco M, Russo S. An architecture for providing java applications with indoor and outdoor hybrid location sensing. *Proceedings of IEEE International Workshop on Future Embedded and Ubiquitous Systems (SEUS 2006)*. IEEE Computer Society Press: Silver Spring, MD, April 2006; 37–42.
- [40] di Flora C, Ficco M, Russo S, Vecchio V. Indoor and outdoor location based services for portable wireless devices. *Proceedings of IEEE International Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI 2005)*. IEEE Computer Society Press: Silver Spring, MD, June 2005; 244–250.
- [41] Nord J, Synnes K, Parne P. An architecture for location aware applications. *Proceedings of the 35th Hawaii International Conference on System Sciences*. IEEE Computer Society Press: Silver Spring, MD, 2002; 293–299.
- [42] Hightower J, Brumitt B, Borriello G. The location stack: Layered model for location in ubiquitous computing. *Proceedings of the 4th IEEE International Workshop on Mobile Computing System and Applications*. IEEE Computer Society Press: Silver Spring, MD, 2002.
- [43] A. Harter et al., “The Anatomy of a Context-Aware Application,” *Proc. 5th Ann. Int’l Conf. Mobile Computing and Networking (Mobicom 99)*, ACM Press, New York, 1999, pp. 59-68.
- [44] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-Support System,” *Proc. 6th Ann. Int’l Conf. Mobile Computing and Networking (Mobicom 00)*, ACM Press, New York, 2000, pp. 32-43.
- [45] Philipp B, Redpin - adaptive, zero-configuration indoor localization through user collaboration, presented at the Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, San Francisco, California, USA, 2008.
- [46] Lin H, Zhang Y, Griss M and Landa I, Enhanced Indoor Locationing in a Congested Wi-Fi Environment, 2009; MRC-TR-2009-04).
- [47] Bill NS, Anthony L, Gaetano B, William GG, David M, Edward L, Anand B, Jason H and Vaughn I, Challenge: ubiquitous location-aware computing and the "place lab" initiative, presented at the Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots, San Diego, CA, USA, 2003.
- [48] LaMarca A, Chawathe Y, Consolvo S, Hightower J, Smith I, Scott J, Sohn T, Howard J, Hughes J, Potter F, Tabert J, Powledge P, Borriello G and Schilit B, Place Lab: Device Positioning Using Radio Beacons in the Wild. *Pervasive Computing*, 2005, pp. 116-133.
- [49] Timothy S, William GG, James S, Anthony L, Yatin C, Ian S and Mike C, Experiences with place lab: an open source toolkit for location-aware computing, presented at the Proceedings of the 28th international conference on Software engineering, Shanghai, China, 2006.

- [50] Hightower J, LaMarca A and Smith IE, Practical Lessons from Place Lab, Pervasive Computing, IEEE, 2006; 5(3), pp. 32-39.
- [51] A. K. Dey, "Understanding and Using Context," Personal and Ubiquitous Computing, vol. 5, pp. 4-7, 2001.
- [52] A. Zimmermann, A. Lorenz, and R. Oppermann, "An Operational Definition of Context," in Modeling and Using Context, 2007, pp. 558-571.
- [53] N. O'Connor, R. Cunningham, and V. Cahill, "Self-Adapting Context Definition," in Self-Adaptive and Self-Organizing Systems, 2007. SASO '07. First International Conference on, 2007, pp. 336-339.
- [54] P. Gray and D. Salber, "Modelling and Using Sensed Context Information in the Design of Interactive Applications," in Engineering for Human-Computer Interaction, 2001, pp. 317-335.
- [55] N. O'Connor, R. Cunningham, and V. Cahill, "Self-Adapting Context Definition," in Self-Adaptive and Self-Organizing Systems, 2007. SASO '07. First International Conference on, 2007, pp. 336-339.
- [56] A. K. Dey, "Providing Architectural Support for Building Context-Aware Applications," Georgia Institute of Technology, 2000.
- [57] G. Broll, H. Hubmann, G. N. Prezerakos, G. Kapitsaki, and S. Salsano, "Modeling Context Information for Realizing Simple Mobile Services," in Mobile and Wireless Communications Summit, 2007. 16th IST, 2007, pp. 1-5.
- [58] C. Stefano, D. Florian, M. Maristella, and M. F. Federico, "Model-driven development of context-aware Web applications," ACM Trans. Internet Technol., vol. 7, p. 2, 2007.
- [59] K. Henriksen, J. Indulska, and A. Rakotonirainy, "Modeling Context Information in Pervasive Computing Systems," in Pervasive Computing, 2002, pp. 79-117.
- [60] V. Bellotti and K. Edwards, "Intelligibility and accountability: Human considerations in context-aware systems," Human-computer interaction, vol. 16, pp. 193-212, 2001.
- [61] P. Gray and D. Salber, "Modelling and Using Sensed Context Information in the Design of Interactive Applications," in Engineering for Human-Computer Interaction, 2001, pp. 317-335.
- [62] T. Buchholz, A. Kupper, and M. Schiffers, "Quality of context: What it is and why we need it," in Proceedings of the 10th HP-OVUA Workshop, 2003.
- [63] M. Krause and I. Hochstatter, "Challenges in Modelling and Using Quality of Context (QoC)," in Mobility Aware Technologies and Applications, 2005, pp. 324-333.
- [64] T. Zimmer, "Qoc: Quality of context-improving the performance of context-aware applications," Adjuncted Proceedings of Pervasive 2006, 2006.
- [65] K. Sheikh, M. Wegdam, and M. van Sinderen, "Middleware Support for Quality of Context in Pervasive Context-Aware Systems," in Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on, 2007, pp. 461-466.
- [66] M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyrtatos, "Contextualization as an Abstraction Mechanism for Conceptual Modelling," in Conceptual Modeling — ER '99, 1999, pp. 77-77.
- [67] M. Theodorakis, "Contextualization: An Abstraction Mechanism for Information Modeling," in Department of Computer Science: University of Crete, Greece, 1999.
- [68] I. Hochstatter, A. Küpper, M. Schiffers, and L. Köthner, "Context Provisioning in Cellular Networks," in 8th International Workshop on Mobile Multimedia Communications Munich, Germany, 2003.

- [69] F. Kilander, W. Li, C. G. Jansson, T. Kanter, and G. Maguire, "Distributed Context Data Management," in *Proceedings of the First International Workshop on Managing Context Information in Mobile and Pervasive Environments* Ayia Napa, Cyprus 2005.
- [70] B. M. Eshete, "Context Information Refinement for Pervasive Medical Systems," Addis Ababa University, 2007.
- [71] K. Henriksen, J. Indulska, and A. Rakotonirainy, "Modeling Context Information in Pervasive Computing Systems," in *Pervasive Computing*, 2002, pp. 79-117.
- [72] M. Krause and I. Hochstatter, "Challenges in Modelling and Using Quality of Context (QoC)," in *Mobility Aware Technologies and Applications*, 2005, pp. 324-333.
- [73] S. Amundsen and F. Eliassen, "A resource and context model for mobile middleware," *Personal and Ubiquitous Computing*, vol. 12, pp. 143-153, 2008.
- [74] M. A. Razzaque, S. Dobson, and P. Nixon, "Categorization and Modelling of Quality in Context Information," in *Proceedings of the IJCAI 2005 Workshop on AI and Autonomic Communications*, Edinburgh, Scotland, 2005.
- [75] C. Anagnostopoulos, A. Tsounis, and S. Hadjiefthymiades, "Context Awareness in Mobile Computing Environments," *Wireless Personal Communications*, vol. 42, pp. 445-464, 2007.
- [76] C. Stefano, D. Florian, M. Maristella, and M. F. Federico, "Model-driven development of context-aware Web applications," *ACM Trans. Internet Technol.*, vol. 7, p. 2, 2007.
- [77] J. Daehyung, L. Jeongkeun, L. Semun, H. Taejoon, K. Taekyoung and C. Yanghee, Signal Dragging: Effects of Terminal Movement on Wardriving in CDMA/WCDMA Networks, *Lecture Notes in Computer Science: Location- and Context-Awareness*, (4718), 2007, 211-227.
- [78] H. Jinyoung, L. Jeongkeun, T. Kwon, J. Daehyung, H. Taejoon, C. Yanghee, How to Mitigate Signal Dragging during Wardriving, *IEEE Pervasive Computing*, 9(1), 2010, 20-27.
- [79] Kuo H, *White noise distribution theory*. CRC Press, 1996.
- [80] Chhabra S, Zaghloul A and Kilic O, Co-Channel Interference in Satellite-Based Cellular Communication Systems, 2005.
- [81] Chuan-Chin P and Wan-Young C, Mitigation of Multipath Fading Effects to Improve Indoor RSSI Performance, *Sensors Journal, IEEE*, 2008; 8(11), pp. 1884-1886.
- [82] Cheng C, Hsiao P, Kung H and Vlah D, Adjacent channel interference in dual-radio 802.11 a nodes and its impact on multi-hop networking, *IEEE Globecom 2006*, Citeseer, 2006, pp. 1-6.
- [83] Ekahau HeatMapper. <http://www.ekahau.com/products/heatmapper/overview.html>
- [84] Yan Qiu C, Damper R.I and Nixon M.S, On neural-network implementations of k-nearest neighbor pattern classifiers, *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on* , vol.44, no.7, 1997, pp.622-629.
- [85] Tran Q, Tantra J, Foh C, Tan A, Yow K and Qiu D, Wireless Indoor Positioning System with Enhanced Nearest Neighbors in Signal Space Algorithm, *IEEE Vehicular Technology Conference, 2006. VTC-2006 Fall*, 2006, pp. 1-5.
- [86] Media P, Pinch Media – Mobile Application Analytics – iPhone Analytics; 29 July 2009). Available at: <http://www.pinchmedia.com/appstore-secrets/>.
- [87] R. Rogers, J. Lombardo, Z. Mednieks & B. Meike, *Android Application Development: Programming with the Google SDK*, O'Reilly Media, Inc, 2009.
- [88] G. Chang, C. Tan, G. Li & C. Zhu, Developing Mobile Applications on the Android Platform, *Mobile Multimedia Processing, Lecture Notes in Computer Science*, 2010, 264-286.

- [89] S. Xianhua, D. Zhenjun & C. Rong, Research on Mobile Location Service Design Based on Android, *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, 2009, 1-4.
- [90] P. Heming, J. Linying, Y. Liu & Y. Kun, Research of android smart phone surveillance system, *Computer Design and Applications (ICCD A), 2010 International Conference on*, 2010, 373-376.