Enabling Adaptive Context Views for Mobile Applications

Negotiating Global and Dynamic Sensor Information

Stefan Forsström

Mittuniversitetet
MID SWEDEN UNIVERSITY

Department of Information Technology and Media
Mid Sweden University

Licentiate Thesis No. 58
Sundsvall, Sweden
2011

Mittuniversitetet
Informationsteknologi och medier
SE-851 70 Sundsvall
SWEDEN

ISBN 978-91-86694-36-4
ISSN 1652-8948

Akademisk avhandling som med tillstånd av Mittuniversitetet framlägges till offentlig granskning för avläggande av teknologie licentiatexamen fredagen den 29 april 2011 i O102, Mittuniversitetet, Holmgatan 10, Sundsvall.

Tryck: Tryckeriet Mittuniversitetet

The only difference between men and boys,
is the price of their toys

Abstract

Mobile devices with Internet access and large amounts of sensors, pushes the development of intelligent services towards new forms of pervasive applications. These applications are made context-aware by utilizing information from sensors and hence the context of a situation, in order to provide a better service. Based on this, the focus of this thesis is on the challenge of creating context awareness in mobile applications. That both utilizes dynamic context information from globally available sensors and provides adaptive views of relevant context information to applications.

The first challenge is to identify the properties of an architecture that provides scalable access to information from global sensors within bounded time, because existing systems do not support these properties in a satisfactory manner. The majority of related systems employ a centralized approach with limited support for global sensor information due to poor scalability. Therefore, this thesis proposes a distributed architecture capable of exchanging context between users and entities on a peer-to-peer overlay. Pervasive applications can thus utilize global sensor information in a scalable and manageable way within predictable time bounds.

The second challenge to support continually changing and evolving context information, while providing it as both adaptive and manageable views to applications. To address this particular problem, this thesis proposes the usage of a locally stored evolving context object called a context schema. In detail, this schema contains all context information that is considered as being relevant for a specific user or entity. Furthermore, this thesis proposes an application interface that can provide snapshots of the evolving context schemas as adaptive views. These views can then be used in context-aware mobile applications, without inducing unnecessary delays.

By successfully addressing the challenges, this thesis enables the creation of pervasive and adaptive applications that utilize evolving context in mobile environments. These capabilities are made possible by enabling access to global sensor information based on a distributed context exchange overlay, in combination with evolving context schemas offered as views through an application interface. In support of these claims, this thesis has developed numerous proof-of-concept applications and prototypes to verify the approach. Hence, this thesis concludes that the proposed approach with evolving context information has the ability to scale in a satisfactory manner and also has the ability to dynamically offer relevant views to applications in a manageable way.

Acknowledgements

Firsty, I would like to thank my supervisor Prof. Theo Kanter and my secondary supervisor Dr. Stefan Pettersson, for providing me with the guidance and opportunity to make this thesis possible. I have always been interested in state of the art applications, and they have enabled me to pursue research in this field. I would also like to thank the other members of the MediaSense group for their valuable input and discussions. Based on the amount of heart and soul we put into our research, it is of no great surprise that we create amazing things together.

Many thanks also go out to all my colleagues at the department of Information and Communication Systems at Mid Sweden University, who have provided many interesting discussions during lunch and coffee breaks. The same gratitude goes to my colleagues at Clas Ohlson who have made the funding for my undergraduate studies possible, in addition to providing me with a unique insight into peoples everyday problems.

Lastly, but by no means least, I would like to thank all my family and friends for their love and encouragement. You might not fully understand the research I work with, but I could never have done this without your help and support.

Table of Contents

List of Papers

This thesis is mainly based on the following ten papers, which can be found in the reference list as references [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. But they will be herein referred to by their Roman numerals.

I T. Kanter, S. Pettersson, S. Forsström, V. Kardeby, and P. Österberg. "Ubiquitous Mobile Awareness from Sensor Networks". In *Mobile Wireless Middleware, Operating Systems, and Applications-Workshops. Springer, 2009*, pp. 147-150.

II T. Kanter, P. Österberg, J. Walters, V. Kardeby, S. Forsström, and S. Pettersson. "The MediaSense Framework". In *Proceedings of 4th International Conference on Digital Telecommunications. IEEE, 2009*, pp. 144-147.

III T. Kanter, S. Forsström, V. Kardeby, J. Walters, P. Österberg, and S. Pettersson. "Ubiquitous Mobile Awareness from Sensor Networks". In *Enabling Context-Aware Web Services. CRC Press, 2010*, ch. 16, pp. 449-463.

IV T. Kanter, S. Pettersson, S. Forsstrom, V. Kardeby, R. Norling, J. Walters, and P. Österberg. "Distributed Context Support for Ubiquitous Mobile Awareness Services". In *Proceedings of 4th International Conference on Communications and Networking in China. IEEE, 2009*.

V S. Forsström, V. Kardeby, J. Walters, R. Norling, and T. Kanter. "Dissemination of Anonymised Context Information by Extending the DCXP Framework". In *Mobile Networks and Management: First International Conference. Springer, 2009*, pp. 57-66.

VI V. Kardeby, S. Forsström, J. Walters, P. Österberg, and T. Kanter. "The Updated MediaSense Framework". In *Proceedings of Fifth International Conference on Digital Telecommunications. IEEE, 2010*, pp. 48-51

VII S. Forsström, V. Kardeby, J. Walters, and T. Kanter. "Location-Based Ubiquitous Context Exchange in Mobile Environments". In *Mobile Networks and Management: Second International Conference. Springer, 2010.*

VIII T. Kanter, V. Kardeby, S. Forsström, J. Walters. "Scenarios, Research Issues, and Architecture for Ubiquitous Sensing". In *Mobile Networks and Management: Second International Conference. Springer, 2010.*

IX S. Forsström and T. Kanter. ”Continuously Evolving Context Information by Utilizing Sensors in Mobile Environments”. Submitted to *1st International Workshop on Opportunistic Sensing and Processing in Mobile Wireless Sensor and Cellular networks. 2011*.

X S. Forsström and T. Kanter. ”Enabling Adaptive Context Views for Mobile Applications utilizing Dynamic Sensor Information”. Submitted to *4th International ICST Conference on Mobile Wireless Middleware, Operating Systems, and Applications. 2011*.

In detail, papers I-II and IV-VIII are peer reviewed and published in international conference proceedings. Paper III is a chapter published in a paper compilation book. Paper IX and paper X are submitted to conferences, awaiting acceptance. Furthermore, the contents of papers I-II, IV-V, and VII are included in the end of this thesis as an extension to chapters 3 and 4.

List of Figures

List of Tables

Terminology

Abbreviations and Acronyms

3G	Third generation cellular wireless standards
4G	Fourth generation cellular wireless standards
API	Application Programming Interface
CCN	Content Centric Networking
CoOL	Context Ontology Language
CUA	Context User Agent
DCXP	Distributed Context eXchange Protocol
DHT	Distributed Hash Table
GPS	Global Positioning System
HSDPA	High-Speed Downlink Packet Access
HTTP	HyperText Transfer Protocol
IMS	IP Multimedia Subsystem
IP	Internet Protocol
JavaME	Java Micro Edition
LTE	3GPP Long Term Evolution
NetInf	Network of Information
OWL	Web Ontology Language
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
UCI	Universal Context Identifier
UDP	User Datagram Protocol
XML	eXtensible Markup Language

Mathematical Notation

C	Context information
$C_{evolving}$	Evolving context information
C_{local}	Local context information
$C_{relevant}$	Relevant context information
C_{remote}	Remote context information
$f_{cp}()$	Context proximity function
d	Context proximity distance
C_{stored}	Stored context information
$D_{application}$	Application delay
$d_{transmission}$	Transmission delay
$d_{central\ database}$	Delay for centralized database processing
$d_{local\ database}$	Delay for local database processing
$D_{propagation}$	Propagation delay
D_{total}	Total delay

Chapter 1

Introduction

Recent advances in mobile technology have enabled the provision of pervasive applications capable of providing different services based on the status of users and entities. This thesis defines such applications as context-aware, because they use sensor technology to gather raw contextual data from the environment. This raw data can then be used to create better applications by involving the situation of the users in the application behavior. This can for example be done by altering the user interface based on the profile of the user, or displaying advertisements based on the current location and time of day. The current trend in mobile devices is moving towards high speed mobile broadband Internet access and multitudes of sensors, which pushes the development further towards these new forms of pervasive applications. Therefore, this thesis explores the challenges that have arisen in the search for adaptive context-aware applications in mobile environments. This includes architectural design for dynamic context exchange, relevance based context data management, and applications access of dynamic context information in a manageable way.

1.1 Background and Problem Motivation

The proliferation of mobile context-aware applications has been held back by underlying systems which do not support continuously changing dynamic context. Adaptive context-aware applications require a flexible underlying architecture, with a natural approach to the dynamic behavior of context information. This means that the context information must be allowed to continuously change and evolve over time. However, it must still be made available to the applications within predictable time bounds, as views of the currently relevant context information.

The concept of context is commonly understood as the situation or surroundings of an entity, where this entity can be any kind of item with an associated situation, such as a user, a device, or an object. However, this indirect notion of the situation is not naturally understandable by computers. Hence, there has been research into context awareness which has resulted in what is seen today as context-aware applications. In detail, these applications utilize sensors to autonomously acquire context about a user or entity, which can be used at a later stage in informed decision making in order to create intelligent applications. Many applications currently exist which can be considered context-aware. These applications are for example, the sorting of search results based on a user's location and adaptable advertisements based on a user's interest. In general, these context based services and context-aware applications gather context about its users pervasively, without any user interactions. The context is usually gathered from sensors, which is placed into centralized database storage, and then utilized by applications directly from the database. Furthermore, this centralized database enables application providers the ability to control and secure the information flow, thus managing all the context information themselves.

These centralized approaches create unnecessary bottlenecks and induce single points of failures. The centralization also induces a vertical and narrow minded approach, where each vendor controls its own set of context information with no interest in sharing this data with other parties. This forms a walled garden, which provides the users with a service, given that the users stay within the boundaries of the system. These walled gardens are also specialized for specific tasks and can therefore only use a fraction of the available context information. Vertical and restricted architectures stifle both innovation and proliferation, even if they initially provide a simple mean of developing context-aware applications.

However, ubiquitous and dynamic context-aware applications demand a horizontal and open environment, which also must have certain properties. For example, rapid dissemination of context within predictable time bounds and good user scalability when the amount of users increase in magnitude. These demands point towards a distributed system and the peer-to-peer paradigm, which provides both better scaling of resources and better support for direct exchange between entities.

1.2 Overall Aim

The aim of this thesis is to contribute methods which enable context awareness within context-aware mobile applications, which can both acquire context from global sensors and exchange the context information between entities. In addition, they must also provide context information to applications as relevant and adaptive views in ubiquitous and global scenarios. Therefore, the focus in this thesis is on scenarios which demand a massive user base and applications which require access to continuously changing context information. This includes scenarios such as the exchange of road characteristics and environmental information between a large number of entities traveling at high speeds during hazardous weather. Another example involves the enabling of context based advertisements when users are near different shopping malls, which can enable them to receive notifications based on the number of available parking spaces, relevant sales based on the current fashion, or the availability of the users clothing size in the store. Furthermore, the scenarios can also include social applications, such as finding other users with similar interests,

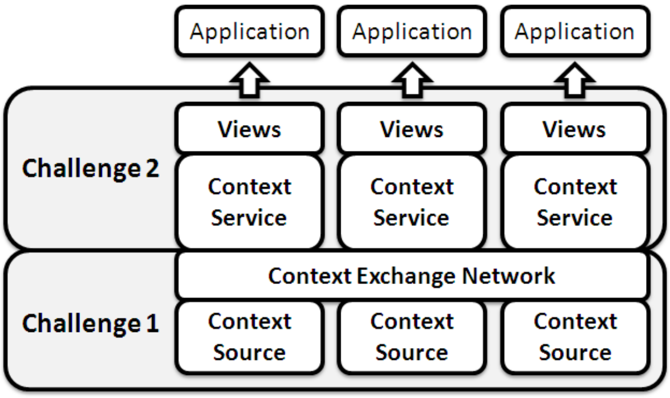

Figure 1.1: Overview of the two challenges.

or providing notifications relating to status changes of relevant entities based on the current situation.

This thesis aims to improve the proliferation and innovation of these adaptive context-aware applications. Some of these context-aware applications already exist, but are limited in regards to their dynamic properties. This must however change for the future Internet of Things where all objects are connected and can communicate, which will require better dynamics and adaptiveness in applications. Therefore, this thesis aims to both enable and simplify the development of dynamic context-aware applications, by exploring methods for providing dynamic context as adaptive views in mobile applications. With this goal in mind, the aim is to create an understanding and a foundation for mobile context-aware applications running in a wider range of scenarios. In addition, the aim is also to enable applications to utilize context information derived from global sensors, in order to display context-aware properties and behavior.

1.3 Scope

This thesis has two main challenges which will be explored. They are in detail, the enabling of global context dissemination between entities, and the enabling of dynamic evolving context information as manageable views in context-aware mobile applications. An overview regarding how these areas fit together can be seen in figure 1.1. In overview, the figure shows a distributed approach providing context information to the applications, by utilizing a distributed context exchange network for exchanging global context information between entities in a scalable manner. To be specific, the context exchange network addresses the first challenge because it can negotiate context information from global context sources, and the distributed context service solves the second challenge because it can provide views of the exchanged context information to applications. However, because this thesis focuses on open dissemination of context between entities, it will not study issues related to authority based access management, such as supervised control of the exchange between entities. Hence, the assumption is that each entity handles its own access management and is in control over its own sensors and devices. Furthermore, no attempt is made to study the different radio technologies used for wide area Internet access and therefore the assumption is that adequate connectivity through mobile broadband packet data access exists, such as 3G (HSDPA) or 4G (LTE).

1.3.1 Supporting Global Context Dissemination

To solve the first challenge relating to the exchange of global context information between users and entities, the dissemination of context information will require a common architecture and approach. So far the usage of proprietary and centralized approaches has been dominant, although an open and distributed approach offers far more benefits especially in regards to scalability. However, creating context awareness from global sensors is still a problem in distributed systems, because of the critical time constraints introduced because of the dynamics of the continually changing context information. Furthermore, the limited resources and unstable connections of mobile devices also introduces additional problems. Therefore, this thesis will explore the challenge of enabling dynamic context awareness from global sensors in current mobile infrastructure. In detail, this thesis will explore middleware approaches and architectures for disseminating global context information between ubiquitous mobile users and entities.

1.3.2 Enabling Dynamic Context Awareness in Mobile Applications

To solve the second challenge in relation to the support of continually changing context information in mobile applications, adaptive application interfaces are required for the management of the dynamic context information. In a manner similar to that of the first challenge, the usage of centralized database interfaces dominates the related work, even if it does not provide satisfactory support for dynamic context. Context-aware mobile applications demand adaptive answers to be received within critical time frames and within predictable time bounds. A common approach to developing these dynamic applications is therefore demanded, which will be addressed in this thesis. In detail, this thesis has explored methods for enabling dynamic context-awareness in mobile applications based on exchanging relevant context information between entities. In addition, it has also explored a snapshot application interface to minimize the delay when accessing the dynamic context information in mobile environments.

1.4 Concrete and Verifiable Goals

The goal of this thesis can be formalized into three concrete goals, which can be derived from the two challenges. In detail, these goals are:

1. Enable global context information from sensors to be made available in mobile applications, by disseminating context information between users and entities. In detail, the context should both be derived from global sources and be utilized in ubiquitous scenarios with large amounts concurrent entities.
2. A method for providing context information as continually changing adaptive views of relevant context information, which can be utilized in mobile applications. In detail, the method should autonomously manage the context information in a manner which represents the continuous change of context, while providing it to applications in a manageable way.
3. A method for handling the propagation of context information, such as controlling the reach of the private context of an entity. In detail, the method should provide manageable means of controlling the information locally by each entity, thus limiting the dissemination of the personal context information of an entity in the system.

1.5 Methodology

Because this thesis focuses on dynamic context and scalability, related centralized approaches have been investigated theoretically in order to disregard them for further investigation and limit the search for a suitable system. Following this assumption, this thesis thus explored distributed approaches, in particular a peer-to-peer overlay platform capable of context exchange between end users and entities. Proof-of-concept applications and prototypes were created for this platform, which validated the approach because they enabled global context dissemination between entities. The proof-of-concept applications and prototypes were also crucial for the later validation of the approach with evolving contextual views, in order to establish completeness and validity with regards to the whole approach. Complex simulations of a massive system with a large number of clients attached via various access networks are considered to be future work. This is due to the lack of support for context information exchange in simulation software, which would require a simulator to be built exclusively for this thesis.

1.6 Contributions

The author's contributions are presented in ten papers, but only papers I, II, IV, V, and VII are included at the end of this thesis, because they contribute directly to the approach used in this thesis. An overview with regards to how all ten papers fit into

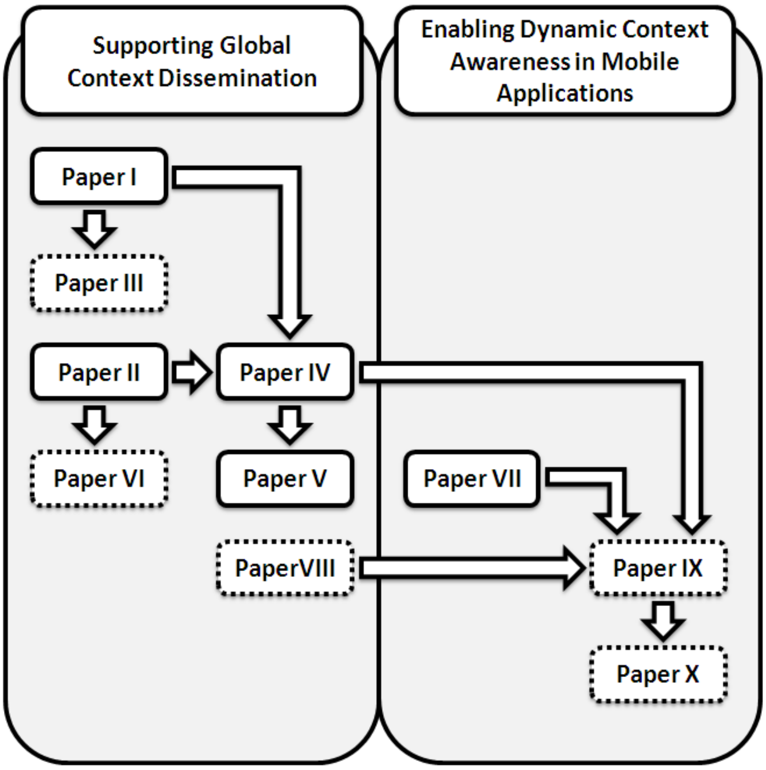

Figure 1.2: Paper contributions.

the two challenges can be seen in figure 1.2. In this figure, the arrows indicate a leads to relation, because most of the papers builds on conclusions from previous results. All of the ten papers are co-authored with fellow researchers within the research group, therefore this section will explain the specific contributions made by the author. The author published seven papers in the area of supporting global context dissemination: Papers I, II, III, IV, V, VI, and VIII. These papers presents solutions for disseminating global context in a distributed manner for a wide range of scenarios. This study resulted in the establishment of different frameworks and possible extensions to these frameworks. The author has also published one paper (paper VII) and submitted two papers (papers IX and X), in the area of enabling dynamic context awareness in mobile applications. These three papers present approaches for enabling relevant context information as adaptive views in mobile applications. In detail, this study resulted in the creation of the evolving context schemas and the snapshot application interface.

Paper I

Ubiquitous Mobile Awareness from Sensor Networks

This paper contributed with an approach to utilizing distributed architectures for creating context information in mobile devices. In addition, a proof-of-concept application was created which demonstrated that the approach is valid. This paper therefore contributed in addressing goal 1, because it created context information from a global sensor. In this paper, the author's contribution related to the validation of the peer-to-peer network. In detail, the author developed and tested the proof-of-concept application. Furthermore, the author also validated the fact that connectivity to related presence architecture was possible.

Paper II

The MediaSense Framework

The contribution of this paper was with an approach to create a peer-to-peer network for exchanging context information between entities. In addition, the paper also contributed to the foundation of the forthcoming papers regarding the formalization of the DCXP protocol. Hence, this paper contributed to goal 1 with a framework design for global context exchange. In this paper, the contribution from the author was with a validation of the approach by means of the proof-of-concept applications based on the peer-to-peer network. In detail, the author verified the proof-of-concept application and developed the connection to the sensor platform.

Paper III

Ubiquitous Mobile Awareness from Sensor Network

This paper contributed further details to the approach with regards to utilizing distributed approaches for creating context information in mobile devices. In addition to this, the proof of concept application was further evaluated and the operation with other existing systems was proven. Because this paper included further details on both the framework and sensor connectivity, it contributed to the addressing of 1. In this paper, the author contributed the idea and development of the DCXP topology with different agents and super nodes. In addition to this, the author also validated the approach with the proof-of-concept application and the sensor connectivity system.

Paper IV

Distributed Context Support for Ubiquitous Mobile Awareness Services

This paper contributed the formalization and implementation of the DCXP protocol, which is capable of disseminating context information between users and entities. In addition, the paper also demonstrated its viability and superiority in comparison to related solutions. This paper contributed to goal 1 since it defines the DCXP protocol in detail and thus enabled global exchange of context information. In this paper, the author's contribution was in relation to the development, formalization, and validation of the DCXP topology with different agents and super nodes, in addition to validating the approach with regards to the developed proof-of-concept application.

Paper V

Dissemination of Anonymised Context Information by Extending the DCXP Framework

The contribution of this paper was an innovative approach to enabling anonymity in context related systems, in particular the DCXP network. In addition, the paper also demonstrated the extensibility of DCXP and peer-to-peer based context exchange, which shows the possibilities of extending these networks to suit the requirements of the users. This paper contributed to the addressing of goal 3, because it provided a manageable approach to managing the reach of an entity's private context. In this paper, the author is one of the original initiators to the anonymity approach with dynamic grouping. Furthermore, the author refined the DCXP messages to fit the current DCXP architecture.

Paper VI

The Updated MediaSense Framework

The contribution of this paper was in relation to an approach aimed at extending the MediaSense framework. In addition, the paper also demonstrated the extensibility in proof-of-concept applications, which further shows that there is possibility of extending the framework to suit the user's needs. This paper partially contributes to goal 1, because it extends the global access of context information further with cloud extensions and home entities. In this paper, the contribution of the author was in relation to the cloud storage idea of context information connected to home gateways. More specifically, the author verified the approach by means of proof-of-concept applications, utilizing sensor platforms connected via mobile phones and home gateways.

Paper VII

Location-Based Ubiquitous Context Exchange in Mobile Environments

A different approach to maintaining a context based network was the contribution of this paper, since it utilized gossiping instead of distributed hash tables. In addition to this, a proof of concept application demonstrated that it is possible to maintain a network of relevant entities. This paper contributed to goal 2 since it provided a new form of method in relation to managing the system, thus providing adapting views of the available data to the applications. In this paper, the author's contribution was by means of the algorithms and their implementations, in addition to the idea of utilizing context proximity. In detail, the algorithms were the achievement of the author and he additionally developed the application, performed the evaluation, and validated the approach.

Paper VIII

Scenarios, Research Issues, and Architecture for Ubiquitous Sensing

This paper contributed a list of research issues for ubiquitous sensing and context-aware applications, in addition to a proposed framework model. In this paper, the contribution of the author was in determining a number of these research issues. This paper thus contributed by determining the issues that are addressed by the three goals in this thesis. The author also contributed with the development of all sensor based components. In detail, the author developed the sensor gateway functionality, the Zigbee sensor connectivity, and the Ethernet sensor connectivity.

Paper IX

Continuously Evolving Context Information by Utilizing Sensors in Mobile Environments

The contribution of this paper was in relation to the ability of creating evolving context in a distributed environment by utilizing agents. But also a realization of an application interface was achieved. This paper therefore contributes to both goal 2 and goal 3, because it enables continuously evolving context views and local management of private context information. In this paper, the author contributed in the idea of having agents evolving context information, stored in a nontraditional database structure. In detail, the author verified the proposed architecture by developing and evaluating the simulation application.

Paper X

Enabling Adaptive Context Views for Mobile Applications utilizing Dynamic Sensor Information

This paper consists of a novel approach to accessing evolving context information as adaptive views. Thus, this paper addresses goal 2 because it provides the means of accessing the adaptive views in a manageable way. The contribution of the author was in relation to both the idea and the achievement of a snapshot application interface, to create adaptive views of continuously evolving context. In detail, the author developed the approach, implemented the application, performed the evaluation, and validated the approach.

1.7 Outline

This thesis is outlined in the following way, chapter 2 includes the background theory in connection to context awareness and related work. Chapter 3 includes the requirements, properties, and how to support global context dissemination. Chapter 4 presents the study into enabling dynamic context awareness in mobile applications. Finally, chapter 5 presents the conclusions and future work.

Chapter 2

Theory and Related Work

In this chapter, the background theory and related work for this thesis is presented. This chapter therefore provides an introduction to context awareness in mobile applications and existing related work. In detail, this chapter includes the definition of context, the definition of context awareness, different context services, and related systems.

2.1 Defining Context

The concept of context is commonly understood as the situation or surroundings of an entity. For example, it is possible to understand the context of a spoken sentence by the situation it is spoken in. Research into the understanding of context by computers has been conducted for some time and one of the earliest accepted definitions of context within computer science was stated by Brown [11]. However Brown was not alone in exploring the area of context, and similar well known definitions are for example by Shilit [12] and by Pascoe [13]. However most current work is based on the view of context provided by the definition given by Dey and Abowd [14], who have refined their definition in stages and finally arrived at *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."*. However, there have also been more recent studies into context, for example by [15], which surveys the area of context and context awareness in detail.

Dey and Abowd [14] do not only define what context is, they also classify and highlight some of the problems associated with context. For example, both the definition of context and all context itself is highly generic, in addition to the fact that context can both be modeled and represented in many different ways. One such example is location, which can be represented both by an address and by a position in latitude and longitude. This opens up a significant problem in relation to context, in that is has to be general and universally transformable between all representations. This modeling of context has been surveyed in detail by [16], in which different approaches to modeling were explored, such as key-value models, markup-scheme models, graphical models, object-oriented models, logic based models, and ontology based models. The survey states that either an object-oriented or an ontology based approach contains the greatest benefits. Examples of outstanding ontology based models are OWL [17] and CoOL [18], and similar outstanding object oriented approaches are explained in [19] and [20].

Context information cannot appear by itself, but must be gathered or acquired from somewhere. Many solutions for this exist, for example some applications use manually oriented approaches such as when a user input personal preferences or a profile. However most solutions today strive towards the detection of context in a more pervasive and automated way, for example by using sensor technology and wireless sensor networks. Because of this, research into sensors and wireless sensor networks is of high importance in relation to the acquisition of contextual information.

2.2 Context Awareness

Context-aware behavior, or more concisely context awareness, is the result which is gained from utilizing context information. In applications this signifies the ability to adapt behavior depending on the current situation of the users. One of the first research applications that has been accredited as being context-aware was the Active badge system by [21]. However, context awareness has since taken significant strides, especially in recent years when there has been an escalation in mobile devices and intelligent applications. Multiple definitions of context awareness exist and context awareness is often synonymous with other terms [14], such as adaptive, reactive, responsive, situated, context-sensitive, and environment directed. One of the first definitions was by [22], but others have also attempted to define it [13, 23]. Dey and Abowd [14] have made a widely accepted and thorough survey in relation to the definition of what context-aware computing is today, and they define context awareness as *"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the users task."*. From these definitions it is possible to deduce that context-aware computing is built on intelligent applications, which can change their behavior according to context. One of the major reasons for the advancement in context awareness during the last few years can be accredited to the mobile platforms, which can travel with the user and become truly pervasive. Many solutions have existed that illustrated context awareness via mobile devices, ranging from the early work of Brown [11], to applications using single context sources such as GPS [24], to the applications being released today such as Waze[1], Foursquare[2], Google Latitude[3], Facebook Advertisements[4] which utilize

[1] Waze: www.waze.com
[2] Foursquare: www.foursquare.com/
[3] Google Latitude: www.google.com/latitude
[4] Facebook Advertsiments: www.facebook.com/advertising

multiple context sources, both from sensors, personal profiles, social networks, and available hardware.

2.3 Context Based Proximity

Proximity can be defined as the physical or geographical nearness, a kind of closeness or vicinity. However, with regards to context the desire is to utilize this concept of closeness in order to create a notion of context proximity. This means the proximity between users' context, i.e. not only by location. For example, two people working in the same company but in two different towns are not in geographical proximity to each other, but their context is in proximity since they both have the same employer. Proximity with regards to context was first defined by Holmquist et al. [25] and studied further in [26]. Although at that point it was quite simple and primitive, because it used the context of artifacts for matchmaking and establishing a connection between small embedded devices. However, context proximity will be utilized in this thesis to create relations between users' context. Furthermore, this thesis assumes that it is possible to rank, relate, and determine the proximity between different the context of users or entities. It is however acknowledged, that related research will have problems solving this in the general case and for all different types of context.

2.4 Architectures for Context Exchange

Related approaches provide insights into current trends regarding architectural design for context awareness systems. In their internal components for exchanging context, trends can be discovered with regards to the distribution, such as centralized, semi distributed, fully distributed, and networked approaches. Therefore, these different architectural approaches will be explained in this section.

2.4.1 Centralized Approaches

Many context based services utilize a centralized approach, because of its simple structure and easy implementation. A generalized figure for a typical centralized context exchange system can be seen in figure 2.1. This figure shows sensors denoted by S that are attached to entities which are denoted by E. These entities can thus send sensor data to a centralized storage in which the context information denoted by C is stored. Furthermore, applications can access context information by querying the centralized storage. However, it will not be possible for this centralized structure to scale in an appropriate manner when the number of users increases in magnitude to billions of users and entities. This is because centralized systems create bottlenecks in the communication flow, and because of the unmanageable size of the stored data in a single database. However, the majority of these solutions claim

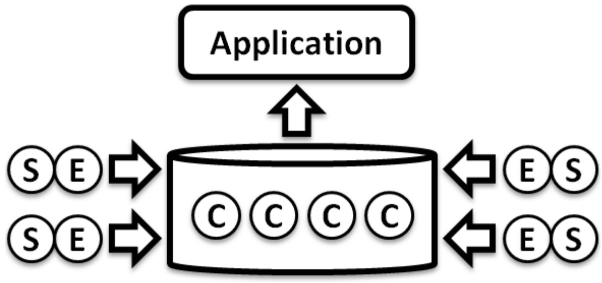

Figure 2.1: Centralized architecture.

that their particular solution scales well even for a large number of users, if provided with sufficient computational power, storage, and replication.

The 3GPP IP Multimedia Subsystem (IMS) [27] and in particular its presence extension (IMS presence) [28] with adaptations for sensing [29], can provide context information to applications. The IMS system approaches context awareness from an operator's point of view, which means it builds a walled garden of distributed entities which stores context centrally in a presence database. This approach offers many benefits from the point of view of the operators, but it does not scale well or support continuously changing context. The SenseWeb project [30] is also a system aimed towards sensors and ubiquitous access to sensor information on the web. The SenseWeb project does however build on centralized web technologies, which provides a traditional centralized approach in relation to exchanging information on the Internet. This approach involving web based storing and dissemination of sensor information is a valid approach for small scale systems, but it will not support a large number of concurrent entities with continuously changing context information. XMPP [31] is also a similar centralized system for both messaging and presence services. XMPP provides context information from centralized databases stored on servers, although replication and clustering of the servers is used in order to provide better scaling in relation to the users.

2.4.2 Semi Distributed Approaches

A structure which is partially distributed, addresses the majority of the scalability problems while retaining the administrative possibilities created by centralization. Semi distributed approaches often use cloud based technologies, replication, and federated brokering to transfer data in a scalable manner. A simplified figure of a general semi distributed system can be seen in figure 2.2. The figure shows that the sensors S and context information C are co-located with the entities E in a distributed manner, hence the exchange can be performed directly between the applications and the entities. However, there is always a governing broker or authority controlling and administrating the setup of the context exchange. Even though the majority of semi distributed systems handle the replication in an efficient manner, they are still not fully distributed and therefore still possess some of the drawbacks

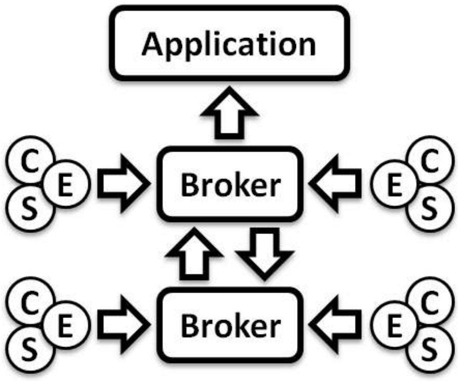

Figure 2.2: Semi distributed architecture.

associated with its centralized components.

The majority of related semi distributed systems build on session establishment protocols such as SIP [32], in order to create the sessions for context exchange between entities. Furthermore, Raz et al. suggested a semi distributed system in [33], which was based on their work during the CONTEXT project. Their solution is typically semi distributed, because it contains peer-to-peer connectivity between the entities but with centralized administrative components. Furthermore, the Mobilife project [34] explored the new possibilities when users' devices become mobile, hence users start living with their mobile devices. The Mobilife project created sample context-aware applications as well as reference architecture for developing context-aware applications. Their system supports context exchange and session establishment, but is also built on partially centralized solutions for managing the network.

The SENSEI project [35] is also built on semi distributed storage of context. SENSEI aims to implement the Internet of Things by means of collaborating wireless sensor and actuator networks with embedded devices. SENSEI can therefore provide network and information management services, which enables reliable context information retrieval and interaction with the environment. These services are for example solved by adding centralized mechanisms for accounting, security, privacy, and trust. However, the SENSEI project possesses the same difficulties in relation to dealing with continuously changing data from a large number of entities, because of the partial centralization.

2.4.3 Fully Distributed Approaches

A fully distributed system has no central point of failure. Therefore, such solutions are naturally more scalable than centralized solutions, but at the cost of communication overhead and computational power on end devices. Context-aware systems building on fully distributed solutions are often built on the peer-to-peer paradigm and file sharing techniques, because of their efficient dissemination of data. Hence,

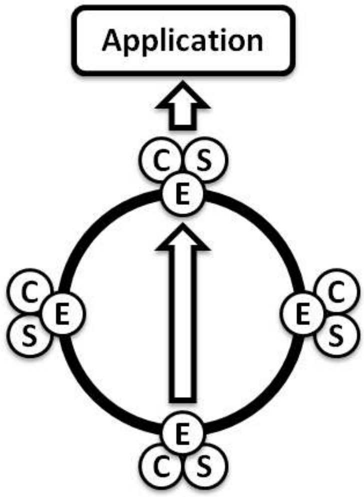

Figure 2.3: Fully distributed architecture.

this approach also supports context exchange and dissemination of context information. However, the distribution creates new problems such as security issues, administration, data management, and overhead. An example of a simplified distributed context exchange system can be seen in figure 2.3. The figure shows that the sensors S and context information C are co-located with the entities E, to create a distributed node. These nodes are then connected through a system which has no central authority and thus exchanging context in a peer-to-peer manner without any centralized components.

Work conducted by Gellersen [36, 37, 38] provides proof of concept application for mobile devices with a fully distributed system, as well as examples regarding how to integrate them into the everyday lives of the users. Their approach includes how to enable additional sensors in mobile devices and an architecture for context awareness based on multiple small and pervasive sensors built into everyday objects. However, because the focus is on smart objects, Gellersen's system does not provide support for global context dissemination. A more agent based approach to context awareness was conducted by Khedr [39, 40], in which agent behavior were combined with ad hoc context based networks. Khedr proposes the usage of an agent-based context-aware infrastructure with an accompanying ontology, which should enable the possibilities of providing spontaneous applications and context provisioning. However, this particular system does not support global context dissemination, since it is aimed towards an ad hoc sensor composition in local proximity.

The SOFIA architecture offers a scalable middleware approach to context aware applications [41]. It can for example, create a smart space which can be utilized by context aware applications. In detail, the SOFIA approach is based on an ontological data model, which can provide the filtering of information in relation to

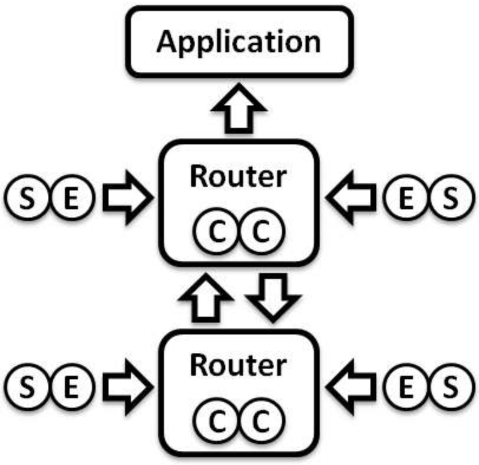

Figure 2.4: Networked architecture.

related context which can be utilized to create context awareness. Furthermore, the COSMOS system is also a middleware for context-centric access control for wireless architectures [42]. COSMOS can dynamically determine the context of mobile users, by taking into account different types of metadata expressed at a high level of abstraction. In detail, the COSMOS system applies this knowledge to create a novel security model for context-centric access control. At a lower level which is closer to the actual sensors, mobile peer-to-peer sensor overlay networks [43] provide a network in which each sensor in the network acts as one peer node. The peers could then use mobile networks to communicate and collaborate on different tasks, or provide information to other sensor nodes that would otherwise not be available.

2.4.4 Network Approaches

The content centric networking explained in [44], also carries an important relation to context-aware systems. The content centric networks use intents instead of naming in order to acquire data, thus making the network transfer the data based on content instead of the normal praxis of transferring the data based on its name. By adopting this change in mentality, many new optimization possibilities have been explored. An example of a simplified content centric network for context exchange can be seen in figure 2.4. The figure shows that the network itself can assist with the exchange of context. In detail, network routers can intermittently store the context information C from the sensors S and entities E, for better dissemination to applications.

The CCNx project [5] is one such content centric network, although other solutions also exist[45]. Most content centric systems are built on the network layer, but content centric solutions also exist which are based on middleware and overlays. For example, it could be argued that the idea of magnet links in the BitTorrent protocol [46]

[5] Project CCNx: www.ccnx.org

is highly content centric, although it is built as an IP based overlay. Context-aware networking is also an important area which has been studied in order to find methods for context-aware communication. This includes for example network routers that change their behavior depending on the context of the end nodes. For example, work performed in [47] is aimed toward adding this intelligence and self learning to network management, so as to enable automatic reconfiguration of the network based on high level context based requirements. There is also the Ambient Networks project[48], which aims to create composeable overlay networks. Ambient Networks major area is within seamless transfer and composition, which are all properties that are also beneficial for distributed context-aware systems. The Network of Information (NetInf) [49] also follows the content centric approach, but from a different angle. NetInf can be adapted for context awareness, since it utilizes information objects which represent a unit of data. These information objects possess globally unique identifiers and can be exchanged in the system, regardless of the object's physical location. This means that the NetInf approach provides identification and location split that is advantageous for both the dissemination of data and context awareness, because it disconnects the context source from the actual data.

2.5 Chapter Summary

In summary, context-aware systems should be built as a system that can naturally enable and support context-aware applications. This can be seen as a platform for enabling applications based on context in an intuitive and effective way. These context-aware systems have many requirements and characteristics that define the system. Such characteristics are ubiquitousness (pervasive, mobile, and ambient deployment), awareness (both with regards to intelligent applications, network awareness and user friendliness), distributed (but must still maintain control of data and provide stable service), and secure (because of all the private information the network manages). However all of these requirements pose important problems, especially within the area involving the acquisition of context (how to use complex sensors), context modeling (how context should be organized), context interpretation (how to understand the meaning of context), architectural design (how to support global context dissemination with predictable time bounds), and decision making (what to do, based on the context). Furthermore, the architectures providing the greatest number of benefits are fully distributed systems, because they conform in a better manner to the scalability requirements. The distributed architecture does however pose its own problems such as security issues and overhead. However, a general and common context-aware system is required in order to cope with the proliferation of pervasive mobile context-aware applications utilizing and disseminating global context information.

Chapter 3

Supporting Global Context Dissemination

This chapter presents methods for disseminating context between users and entities, but also methods for creating context from global sensor information. The dissemination follows the peer-to-peer paradigm, which means that the context information is exchanged on a direct connection basis between entities. By using this method the context is disseminated in a well scaled manner without multi-hopping, proxies, or centralized components. Therefore, this approach and its validity have been explored in a number of papers whose contributions will be explained in this chapter.

3.1 Creating Context from Global Sensors

With the general definition of context used throughout computer science, it can be interpreted in a somewhat vague manner and can contain almost everything that is connected to an entity's situation. This context definition therefore poses some significant problems for context-aware applications, because it is not possible to acquire all the context information regarding a situation. Generally, context data is gathered from sensors or wireless sensor networks, which sense and gather data from the environment. Furthermore, this sensor based context can also be pervasively gathered over time as the context of the entities change. This automation provides sensor based context with a higher accuracy than manually entered context, such as personal preferences, since sensor based context can be gathered without user interaction.

The number of sensors in devices has increased exponentially throughout the last decade, since almost all devices today possess some sort of in-built sensor. It is however only in recent years that it has become possible to access these sensors through wireless sensor networks and similar architectures. One example of such a device is the Bluetooth sensor gateway, see figure 3.1. This is a sensor that was developed and

Figure 3.1: Bluetooth sensor module.

Figure 3.2: Zigbee sensor module.

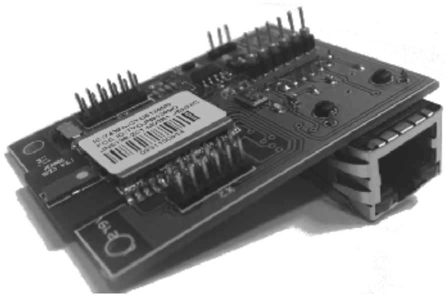

Figure 3.3: Ethernet sensor module.

used in the proof-of-concept applications during this thesis. The sensor can connect via Bluetooth to nearby mobile phones and provide them with raw sensor values. In this case, it was only possible for the sensor to gather temperature and humidity, but with the additional GPS of the mobile phone, intelligent applications could utilize the sensor values in terms of higher meaning context. In detail, the Bluetooth sensor gateway uses a lithium battery as its power source, which means that it will have a long lifetime, but will eventually require charging. Superseding Bluetooth, the Zigbee standard has become a de facto standard for low power wireless sensor networks, one example of such a Zigbee sensor with power sockets and internal sensors can be seen in figure 3.2. This Zigbee sensor was also developed and used in the proof of concept applications during this thesis. In detail, this sensor has both sensors and actuators, which means that it can both control the mounted wall sockets and sense the power consumption of the devices that are plugged into the socket. Furthermore, since this particular sensor has the advantage of being connected to the household electrical sockets, it can run without battery operation. A third type of sensor used for creating context during this thesis was an Ethernet based sensor. This sensor can be seen in figure 3.3 and it is a sensor module that is powered by household electricity and communicates over the Ethernet and IP based infrastructure. This particular Ethernet sensor can only sense temperature and humidity, but its main advantage is that it can communicate over regular IP based networks without any third party involvement. Because of this, it is very self contained and can

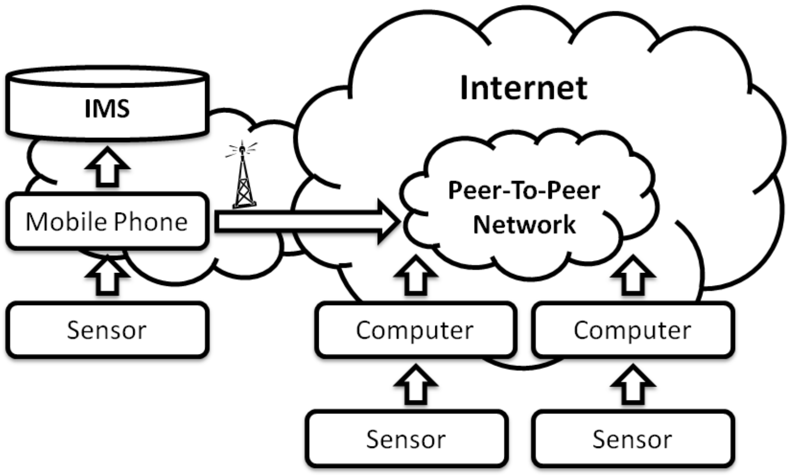

Figure 3.4: The MediaSense framework.

even communicate over the Internet via standard web protocols, such as HTTP.

All these three modules are examples of sensors and actuators that have been used during this thesis to create higher meaning context from raw sensor information. They were also combined with sensors already built into mobile devices, such as GPS and accelerometers. However, a vast number of unexplored sensors and actuators still exist that can be used to create information and context awareness in applications.

3.2 Framework for Global Context Dissemination

Chapter 2 presented related architectures for context awareness and exposed the significant drawbacks of the related approaches. In detail, the majority of these related systems have a centralized or semi distributed approach for context disseminating. Such approaches do not support global context exchange among billions of entities, because it creates unnecessary bottlenecks and unmanageable handling of the data. To address this problem, this thesis has created a new framework for global context dissemination. This framework is built on the peer-to-peer paradigm, which means that the data is transferred directly between the users with minimal delays. This distribution scales better, supports global context dissemination, and supports dynamic context information on a completely different level. To formalize this approach, it was formed into a framework called the MediaSense framework. An overview of this framework can be seen in figure 3.4. The figure shows how mobile phones are connected via wireless sensor networks to acquire context information. This context can then be shared over a global peer-to-peer network. This figure also shows how the framework can operate concurrently with other related context architectures if necessary, in this case the IMS presence.

3.3 The Distributed Context Exchange Protocol

The Distributed Context Exchange Protocol (DCXP) is the peer-to-peer protocol used within the MediaSense framework to exchange context between users and entities. In detail, DCXP is an XML-based application level protocol aimed at providing reliable communication of context among nodes that have joined the overlay network. DCXP imposes a naming scheme similar in format to universal resource identifiers called Universal Context Identifiers (UCIs), which identifies context information stored in the DCXP network. In detail, the UCIs have the following syntax and interpretation:

```
uci://user[:password]@domain[/path[?options]]
```

In this case *uci* is the UCI scheme and *domain* is a fully qualified domain name relating to the location of the context information. The *user* and *password* fields provide an optional means for authorization. *path* adheres to the context information namespace hierarchy used in DCXP, thus allowing organization and sorting of the context in a logical order. *options* facilitates further modifiers in the form of parameter=value. Hence a fully qualified UCI could for example be:

```
uci://alice@miun.se/weather/temp?unit=celsius
```

Thus, these UCIs enables unique identification of context information. Furthermore, with these UCIs it is also possible to relate to context in a formal way which is both human readable and computer understandable.

3.3.1 Protocol Operation

A network that uses DCXP forms a context storage that utilizes a Distributed Hash Table (DHT) to map between UCIs and source addresses. The main advantage in using a DHT is that entries can be found in $log(N)$ time. Thus, the context storage maintains a repository of UCI and source-address pairs which provides a resolving service for the users. DCXP enables the exchange of context between sources and sinks with the use of the distributed lookup system. These sources and sinks are formed and combined in a single endpoint on the end devices, called a Context User Agent (CUA). The CUA corresponds to a node in the DHT and in particular, the CUA enables applications to utilize the DCXP network for disseminating global context information between all entities in the network and within predictable time bounds.

An overview of the DCXP network can be seen in figure 3.5. The ring symbolizes the DHT. Each circle on the DHT symbolizes a node in the DCXP network, and each node has a CUA service. As previously explained, the DHT stores locations for a UCI and location pair, which can be used to establish peer-to-peer connections between end users. DCXP utilizes a SIP inspired protocol with five primitives which the CUAs can utilize. With these five primitives, the DCXP network can establish and exchange context information between the users who have joined the DHT. An example involving the operation of the RESOLVE, GET, and NOTIFY primitives can also be seen in figure 3.5. In this case a CUA resolves a UCI in the network, and at a

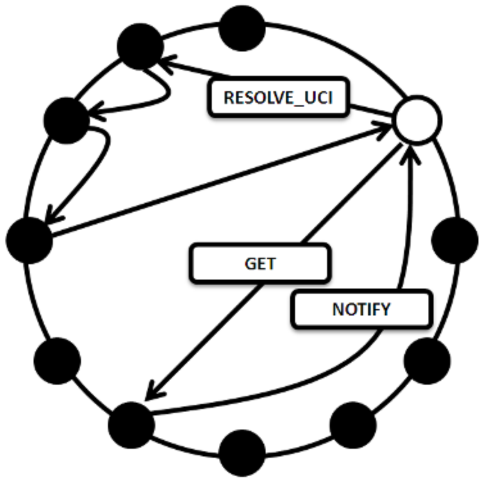

Figure 3.5: The DCXP network and communication.

later stage sends out a GET and receives a NOTIFY with the latest context.

3.3.2 Anonymous Dissemination Extension

In a similar manner to that of the MediaSense framework, the DCXP network was also altered and extended during the work associated with this thesis. One possible extension that was studied for the DCXP network was to enable anonymity for the entities in the system. In order to achieve this, the proposed extension to DCXP employs a hybrid of different anonymity approaches. Firstly, randomly selected users in the DCXP ring are grouped together communicating internally using a token ring based structured. The entities then communicate as a single large anonymous group and because all information is sent and received as a group, individual entities are hidden and remains anonymous. In doing so, a node achieves anonymity by the assumption of ”probable innocence”.

An overview of this architecture can be seen in figure 3.6, which shows groups of nodes and the possibility of exchanging messages among the groups. Each group uses a token ring like protocol to set up a communication structure, such that each node is only aware of its predecessor and successor node of the ring network. This ring network, within each group can be seen in figure 3.7. A token in the ring network carries both encrypted and plain text messages. On receiving a token, the node will attempt to decrypt any encrypted messages with its own private encryption key. All successfully decrypted messages will be acted upon, after which all other encrypted messages are forwarded with the token. Plain text messages are inspected in order to determine whether the contained requests should be executed by this node. However, the plain text messages may entail delivery to the target group of the message instead of forwarding it to the next node in the token ring. This choice is randomly determined and if the current node decides to deliver the message, it will

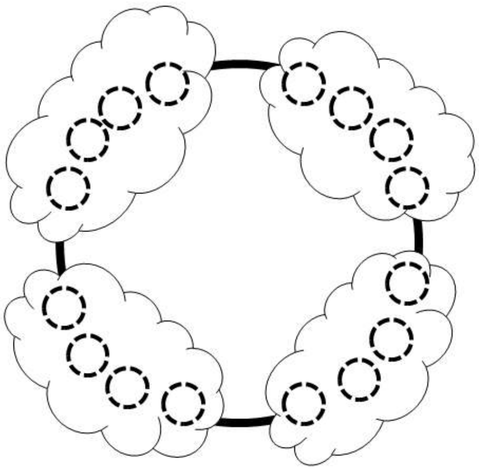

Figure 3.6: The anonymous DCXP network grouping and communication.

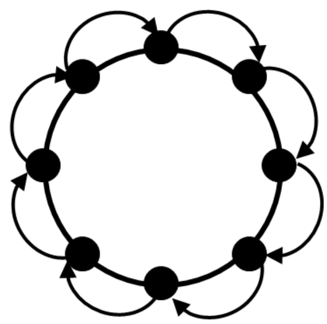

Figure 3.7: The ring network inside each anonymous group.

further inspect and act upon the message depending on the message type. However, if the decision was to not to act on the message then the node delivers the token to its successive node which then undergoes the same decision process. Furthermore, when a node receives a message from an external group, it awaits its turn for the token and delivers the message into the current token. In addition, if a node encounters the same message twice in the token, the message is removed from the token, the assumption being that at this point the message has made a complete round trip and has either been unclaimed or has been acted upon by one of the other peers.

The net outcome of this is twofold, firstly the context source is unable to derive the terminal destination of the requests and responses. Secondly, members of the groups are also unaware of the terminal recipient of the data, since nodes simply forward tokens in a repeated non-discriminatory manner. Furthermore, with this anonymity method the reach of private context information can be managed and controlled by each entity locally, which is required to maintain privacy and management of an entity's private data.

3.4 Chapter Summary

In summary, this chapter presented a method and approach for creating abstract context from raw sensor information. This chapter also presented a distributed framework and protocol for exchanging context information between entities which have joined an overlay network. This approach addresses goal 1 with the dissemination and negotiation of context information via the DCXP protocol in the MediaSense framework. In detail, this approach enables applications to utilize context information derived from global sensors in a fully distributed manner. In comparison to related systems, the presented approach has certain advantages. It is for example fully distributed and has therefore no central point of failure, which is common to all other centralized systems and semi-centralized systems. The fully distributed peer-to-peer exchange also supports a faster exchange of information without proxies, while keeping the delays within predictable time bounds. The presented approach also addresses goal 3, because each entity manages its own context locally and decides whom to share it with. Furthermore, the anonymity enabled DCXP extension provides a means of handling privacy and the reach of the information in a manageable way. Hence, the presented approach can handle the reach of personal context information and control the flow of personal information.

Chapter 4

Enabling Dynamic Context-Awareness in Mobile Applications

This chapter presents a method for enabling dynamic context-awareness by providing adaptive views of continuously changing context information to mobile applications. This method combines the dissemination method explained in the previous chapter together with relevance based context scoping and adaptive application access. Furthermore, by using this method a mobile application can utilize dynamic and continuously changing context information in a scalable manner for a wide range of scenarios.

4.1 Managing Evolving Context Information

Context information has historically been stored in centralized databases that can provide auxiliary functions for the data storage, such as authentication, security, privacy, etc. However with the recent advances in global context sharing, such database approaches do not provide support for dynamic and continually changing context information. To address this problem, each of the end users must maintain their own database locally, in order to minimize both the delay and to enable adaptive application access. This thesis therefore suggests a new database structure called context schema. In detail, this schema is an object which represents the complete contextual knowledge of a particular entity. This schema includes all possible context of that entity, including sensors information, personal profile, preferences, status, etc. Furthermore, because the context is always changing it can be considered as evolving over time. However, to create such evolving context schemas, relations between these context schemas are going to be required. To address this, the system utilizes the concept of context proximity to create relations between entities based on relevance. This means that it is possible to determine that two entities are connected and relevant to each other's schemas, based on the fact that their context information relates to each other by context proximity. In detail, these context schemas can be illustrated algebraically as in equations 4.1, 4.2, 4.3, 4.4 and 4.5. In these equations, all context information C exists in the domain of available context. $C_{evolving}$ is the set of currently evolving context information for a particular entity. This evolving context schema is defined as the union of the set of all local context information C_{local} and the set of all relevant remote context information $C_{relevant}$. Furthermore, this relevant set is determined by $f_{cp}()$, a context proximity function that can determine whether the remote context information C_{remote} is relevant, based on a specified proximity distance d.

$$\forall\, C \in \text{domain of available context} \tag{4.1}$$

$$\{C_{evolving}\} = \{C_{local}\} \cup \{C_{relevant}\} \tag{4.2}$$

$$\{C_{relevant}\} = \{C_{remote} : f_{cp}(C_{remote}) \leq d\} \tag{4.3}$$

$$f_{cp}(\,) = \text{context proximity function} \tag{4.4}$$

$$d = \text{specified proximity distance} \tag{4.5}$$

These context schemas will have to be continually evolved by the system as the context of all entities is constantly changing. This is in contrast to related systems where context is considered as being a static element and thus only a very limited amount of change is allowed. But this thesis foresees that applications might also want to utilize dynamic context, thus our system is made capable of handling evolving context schemas.

4.2 Creating Evolving Context Schemas

To address the problem of evolving context schemas in a distributed manner, this thesis presents an architecture that builds on different types of agents that can enable context schemas to evolve. An overview of our proposed architecture can be seen in figure 4.1. Each of these parts will be explained in greater detail regarding how they contribute to the creation and utilization of evolving context information. But in overview, our architecture contains the evolving context schema, the local gathering agents, the traversing agents, the remote gathering agents, and the context exchange overlay network.

4.2.1 Local Context Gathering Agents

The local context gathering agents are used to gather context from local sources, thus providing the C_{local} set in equation 4.2. These agents are required to create evolving context since most context information has a local sensor as its originating context source of contextual data. These sensors do however introduce their own problems,

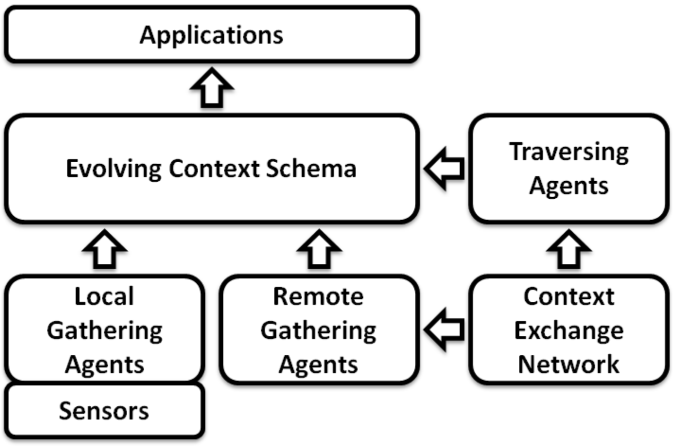

Figure 4.1: Architecture for evolving context.

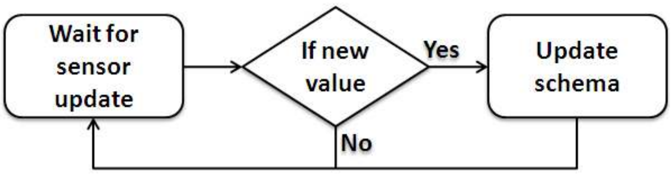

Figure 4.2: Algorithm for the local context gathering agents.

since they only provide raw values and in many cases this is in different formats based on each manufacturer. The problem that each local gathering agent solves is therefore to create context information from many different types of sensors, while providing this context upwards to the evolving context schema. A flow chart of the operation involving the local context gathering agents can be seen in figure 4.2. In detail, the local context gathering agents continually read values from sensors either locally attached, built-in, or connected through local wired and wireless networks. The values gathered from the local gathering agents are then updated into the evolving context schema. Furthermore, pseudo code for the local context gathering agents operation can be seen below.

loop
    wait for sensor update
    if *new sensor value* **then**
        update context schema
    end if
end loop

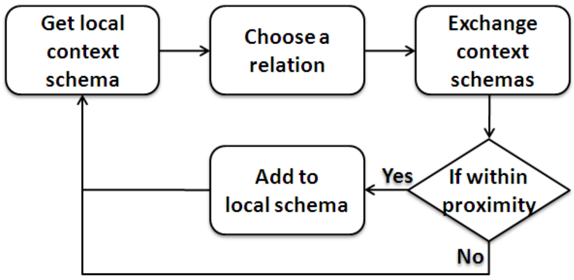

Figure 4.3: Algorithm for the traversing agents.

4.2.2 Traversing Agents

The traversing agents solve the problem of finding new relevant context, thus providing the $C_{relevant}$ set in equation 4.3. In detail, the traversing agent browses the local context schema for new relations that might prove relevant to explore. To find a new user, the traversing agent communicates with a known user from the local context schema, asking for other relevant users. To determine whether another user is relevant, the traversing agent utilizes the context proximity function with a predefined distance. If two users are in context proximity to each other, they are considered relevant and the other user's relevant context is inserted into the local context schema. This algorithm for the traversing agents can be seen in figure 4.3. Because of all the traversing, these agents operate on a best effort system, always traversing the network and evaluating other the context of other users. Each user can have multiple traversing agents operating at the same time, because they can concurrently expand the evolving context schema without internal interference. The basic algorithm for the traversing agent is a five step process. Firstly, it acquires the local context schema. Then it examines the local context schema and chooses a relation which it wants to traverse. Thirdly it traverses this relation, communicating and retrieving the remote user's schema. After it has both the local and the remote schema, it performs an evaluation based on the context proximity between the schemas. Depending on the result of the context proximity evaluation, it determines which parts of the remote context information should be included in the local context schema. Pseudo code for the operation of the traversing agents can be seen below.

```
loop
  get local context schema
  choose a relation
  acquire remote context schema from related user
  for all context in remote schema do
    if is within context proximity then
      add to local context schema
    end if
  end for
end loop
```

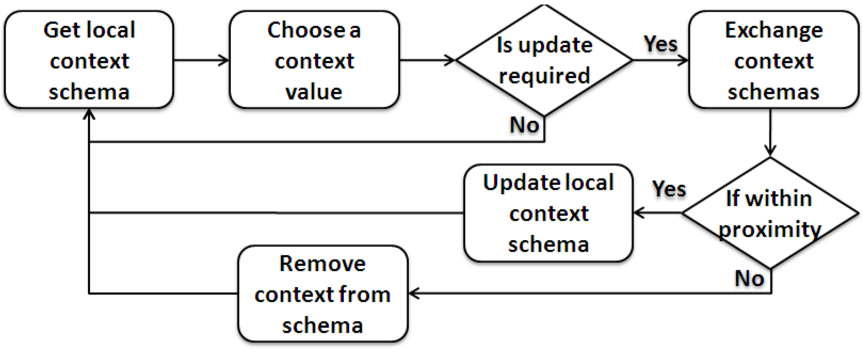

Figure 4.4: Algorithm for the remote context gathering agents.

4.2.3 Remote Context Gathering Agents

The remote context gathering agents operate in a similar manner to that of the local context gathering agents. The main difference is that the local gatherer acquires context from local sources, and the remote gatherer acquires context from remote sources. Therefore the remote gathering agents are responsible for keeping the $C_{relevant}$ set in equation 4.2, continuously updated and accurate. The remote context gathering agents are required because the traversing agents only find new relations, they do not keep the context values continuously updated. In detail, the remote context gathering agents examine the local schema and determines whether a context value requires updating. After this, it establishes a connection to the remote source, acquires the most recent value and updates the local schema. The algorithm for the remote context gathering agent can be seen in figure 4.4. The system will require multiple remote gathering agents, in order to maintain updated context values. Therefore, multiple remote context gatherer agents will run concurrently and acquire context from many different sources at the same time. Furthermore, the pseudo code for the remote gathering agents can be seen below.

loop
  $get\ local\ context\ schema$
  $choose\ a\ context\ value$
  if $value\ require\ update$ **then**
    $acquire\ remote\ context\ schema\ for\ value$
    if $new\ value\ is\ within\ context\ proximity$ **then**
      $update\ local\ context\ with\ new\ value$
    else
      $remove\ context\ value\ from\ local\ context\ schema$
    end if
  end if
end loop

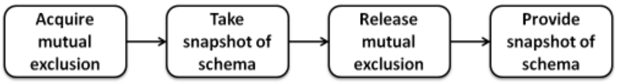

Figure 4.5: Algorithm for the snapshot operation.

4.2.4 Context Information Exchange

The communication performed by all the agents must be conducted in a scalable and dynamic manner, without inducing unnecessary delay. These requirements are supported by the DCXP network. In detail, DCXP provides good scaling because of the logarithmic lookup in its distributed hash table and it also enables the demanded direct dissemination of context between all entities on the overlay. In addition, because the actual dissemination is performed on a peer-to-peer basis without proxies, network delay is kept to a minimum. Furthermore, the DCXP network can run without relying on centralized naming services such as DNS, in addition to providing the option to perform open ended searches utilizing the relations between users on the overlay.

4.3 Providing Adaptive Context Views

The dynamic nature of context and the nature from which context is created, the application interfaces for utilizing context in applications must also support evolving context. This originates from the fact that as the context is always changing, it can become invalid even before it arrives at the application. For example, context can change between the instant it is retrieved from the source and the instant it is received at the application. Hence, this thesis proposes that the application interface should contain a specialized function which applications are able to use in order to interact with the context services which adapts itself as the context changes. This function call should return a snapshot of the local context schema, thus providing a stable copy of the unstable and evolving context. Given that the agents have been successfully evolving the schema, the interface should provide a valid representation of the actual world at the point in time which the snapshot was created. This snapshot can then be searched in at a later stage and used as an adaptive database view in applications. Because of this, the application interface is optimized both with regards to computational costs and delays, since the snapshot operation is very fast, simple and efficient. The simplicity of the snapshot interface can be seen as a flow chart in figure 4.5, which shows that mutual exclusion is acquired for a short period of time while performing the copy operation. After this, the snapshot is returned to the application for processing. This can also be shown in pseudo code, which can be seen below.

lock context schema
copy context schema to snapshot
unlock context schema
return snapshot

4.4 Proof-of-Concept Applications

This approach of creating views of evolving context from global sensors has been implemented in numerous context-aware applications during this thesis. They were created as proof-of-concepts, in order to validate the approach and architecture design. One of the implemented proof-of-concept applications can be seen in figure 4.6. This application was created for low end phones running JavaME, and it utilized the inbuilt GPS and the external Bluetooth sensor. The application shows all users in the system as pins on a map. Each user's sensor information is also displayed, so that the context from the sensor values can be created. In detail, all users are connected to a context service using DCXP, which could negotiate the context data between all users. The result of this is a continuously updating view of all sensor values on a map. Following this proof-of-concept, a follow up and similar application have been created but with a different approach to the dissemination of context. In detail, this proof of concept used the initial ideas to apply context proximity to determine relevant context. A figure of the running application can be seen in figure 4.7. This application was created for an Android mobile phone, and it displays a logical view with the location of all possible relevant entities. However, where the previous application built on a distributed context service using DCXP, this application is built on an ad hoc agent based context exchange which has utilized context proximity to determine relevance.

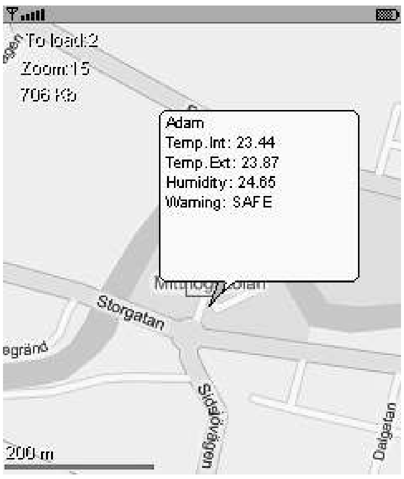

Figure 4.6: JavaME based proof of concept.

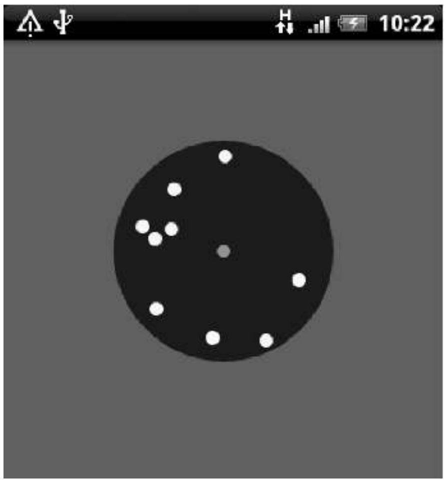

Figure 4.7: Android based proof of concept.

Extending both the previous applications, a third simulation application was created. The focus of this application was on adapting the previous proof-of-concept applications in order to implement the architecture for evolving context schemas. Graphically it is similar to the Android application, since it follows the same agent based approach. Figure 4.8 presents the running simulation, and the figure shows a graphical snapshot view for a particular user's evolving schema. The graphical

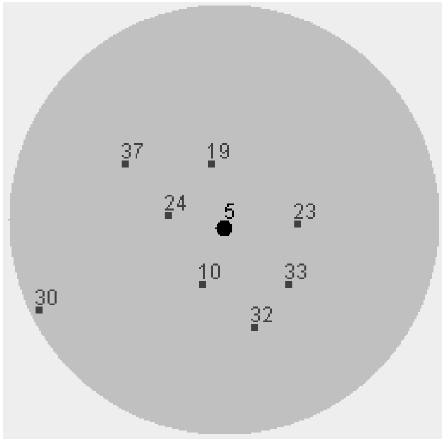

Figure 4.8: A snapshot view from the evolving context simulation.

view is centered on the user, which in this case is user number five. The view also contains the other users who are in context proximity and the predefined context proximity distance can be seen as a circle in the background. This view is updated continuously as the users move around and as new entities enters or leaves the context proximity area. The schema in the simulation does however only contain two dimensional context values, representing for example latitude and longitude. However, in conclusion the simulation proves the feasibility of the approach because it creates views of the evolving context information from negotiated global sensors.

The proof of concept applications was also measured in a preliminary study of the scalability and responsiveness of the different components. In overview, the DCXP network, the agent communication, and the snapshot interface were measured. The first measurement focused on the DCXP network and the response time of the startup and the two primitive operations GET and SET were measured. The results from this measurement can be seen in table 4.1. In this table, the GET operation was measured to be completed in under 0.2s, even in mobile environments. These values were on par with UDP traffic over mobile broadband, which means that no significant additional delays were introduced by the DCXP network itself. The second measurement was performed on an Android mobile phone, which measured the communication of the ad hoc agents. In detail, the response times for the different parts of the algorithm were measured. The most important result can be seen in table 4.2, which shows the measured time for a peer-to-peer communication and the response time for the bootstrapping procedure. These results show that the centralized bootstrapping was the bottleneck of this particular proof of concept application, although it otherwise follows normal TCP traffic speeds over mobile broadband. The third measurement was performed on the simulation application and measured the snapshot interface. The results from this measurement can be seen in table 4.3 and it show the overhead induced by the application interface. In detail, the snapshot application interface performs well, since it only induced minimal de-

Table 4.1: Response times of DCXP.

	Startup	GET	SET
Computer	0.35s	0.02s	<0.001s
Mobile	7.0s	0.20s	0.10s

Table 4.2: Response times in the Android proof of concept.

	10 nodes	25 nodes	50 nodes
Bootstrap	367ms	387ms	410ms
Peer Communication	310ms	313ms	335ms

Table 4.3: Response times of the snapshot interface.

	100 000 context	1 mill. context	10 mill. context
Snapshot Interface	5.3ms	66ms	780ms

lay even when the amount of total context in the local schema was in the magnitude of 10 million entries. Furthermore, these results show that the snapshot operation itself does not cause any significant overhead for creating the adaptive views of the evolving schemas.

4.5 Evaluation

This section presents a validation and evaluation of the proposed approach, i.e., maintaining evolving schemas and adaptive views containing distributed context information. The approach is thus compared with related work regarding qualitative and quantitative aspects, such as the amount of stored context in the system and propagation delay. In detail, the approach is compared to the general architectures studied in section 2.4 with regards to how well these approaches address the goals of the thesis, which were presented in section 1.4. Hence, the results in this chapter are placed into relation with he goals of the thesis, for further establishing the validity of the approach.

4.5.1 Comparison to Centralized Architectures

Relating to the global dissemination of context information in goal 1, centralized systems gather context information in centralized storages which do not scale well with regards to the amount of stored information and to the constant accessing of data. This scaling problem is apparent when there is an enormous increase in the number of global entities, with constantly changing context. The proposed system does however provide decentralized access of the information, which has better scaling and can manage global sensor information. This can also be proven quantitatively by looking at the sum of all context information which a centralized system must manage in its storage. Equation 4.6 shows the total amount of stored context C_{stored}, which is stored in the central database. This can be compared to equation 4.7, which shows the amount stored on each entity in the proposed system. From this it is possible to deduce that the amount of data in the central database will become unmanageable when the amount of total entities increases in magnitude. The proposed system will however still have a manageable set of stored data, because the stored amount is only based on the relevant number of users, not the total number of users in the system. To be concrete, if a system has in total 100 000 entities, with ten context values each, but only fifty of these entities can be considered relevant for a particular application. The total stored size for a centralized system would then be 1 000 000 entries, compared to 510 entries in the proposed approach.

$$C_{stored} = \sum_{n=0}^{total\ entities} n * C_{remote} \tag{4.6}$$

$$C_{stored} = C_{local} + \sum_{n=0}^{relevant\ entities} n * C_{remote} \tag{4.7}$$

The same problem occurs when comparing the delays for an application querying the stored context information. The delay $D_{application}$ for centralized access can be seen in equation 4.8, where the transmission delay over the Internet $d_{transmission}$ is added twice on top of the database query time $d_{central\ database}$. In the proposed architecture the same application access is performed locally, see equation 4.9. Thus the delay is isolated to the database query inside the local database $d_{local\ database}$, which was also proven to contain a much smaller dataset. Hence, the centralized lookup will always be two transmission delays longer than the delay for the local database, regardless of centralized location and replication.

$$D_{application} = d_{transmission} + d_{central\ database} + d_{transmission} \tag{4.8}$$

$$D_{application} = d_{local\ database} \tag{4.9}$$

Furthermore, the proposed system also has a lower propagation delay $D_{propagation}$ for when the context information changes. This can be seen in equation 4.10 and equation 4.11, which shows the delay from the point when the context is updated to that when it has arrived at the application. In centralized systems it is apparent that the context information must be routed through the centralized point, which induces an additional transmission delay over the Internet in comparison to peer-to-peer dissemination. To be concrete, the propagation delay of a centralized system will always add one additional transmission delay over the Internet, because it has to relay the information.

$$D_{propagation} = d_{transmission} + d_{central\ database} + d_{transmission} \tag{4.10}$$

$$D_{propagation} = d_{transmission} + d_{local\ database} \tag{4.11}$$

Because of this, centralized systems also have problems with goal 2 which are the adaptive views, because they must perform massive queries through a very large datasets in order to provide the same functionality. These queries will also affect the dynamics of the system, because a long query requires a stable dataset while it is being performed. This problem can also be found when studying the required workload by the centralized component compared to the distributed workload of each node in the proposed system. The total workload required to create one view for each end application in a centralized system can be seen in equation 4.12, this can be compared to equation 4.13 which shows the required workload on each end node in the proposed system. However, one important thing to note is that the total workload of the whole proposed system, the sum of the workload of all nodes, will be equal to the workload of the centralized workload.

$$Workload_{central} = \sum_{n=0}^{total\ entities} n * (\sum_{m=0}^{related\ entities} m * C_{remote}) \qquad (4.12)$$

$$Workload_{per\ node} = \sum_{n=0}^{related\ entities} n * C_{remote} \qquad (4.13)$$

Centralized systems do however have an advantage over the management and reach of the private context in goal 3, because they have complete control over the flow of information. However, this centralized management will have negative effect on the scalability, since all decisions on the reach of information must be made by the centralized authority.

Both the IMS system, XMPP, and SenseWeb employ centralized storage of context information. Hence they scale accordingly, with respect to equations 4.6, 4.8, 4.10, and 4.12. In conclusion, the proposed solution in this thesis is superior to these related systems, especially in regard of the amount of context information which is stored, the delay when acquiring context information, and the propagation delay when an entities context is updated.

4.5.2 Comparison to Semi Distributed Architectures

The semi distributed approaches studied in 2.4.2 addresses some of the scalability problems associated with centralized solutions, but not all of them because they still maintain some centralized parts. They can address the support of global context dissemination in goal 1 much better than centralized systems, because the actual context exchange is performed outside of the federated broker. In detail, both distributed and semi distributed systems have the propagation delay shown in equation 4.11. However, since a federated broker still becomes a centralized component, it will scale poorly when performing queries on the whole dataset. Thus, semi distributed systems scales as a centralized system, as in equation 4.8 for such actions. Furthermore, because the centralized component will require complete knowledge of the system to provide this service, the system will have to store both centrally in the central component according to equation 4.6 and remotely on each entity as in equation 4.7. Creating a very large total amount of stored context information in the system. Therefore, the proposed system with a distributed approach is still able to offer a better scaling system. In relation to goal 2 which addresses adaptive views, semi distributed systems are also limited by their centralized component. This is because it will require complete knowledge of the data, whose size increases according to equation 4.6. Furthermore, the centralized component has to search through to this large database set to create the relevant view to applications. However, if these drawbacks are disregarded, the gain associated with having centralized components is that they can still maintain the flow of information as a centralized system which is a demand of goal 3.

The federated brokers of SENSEI and the SIP enabled context exchange explored in the CONTEXT and Mobilife projects, offers significantly more scalable approaches to sharing context than centralized systems. Thus scaling according to equation 4.11 for simple context exchange between entities. But for system wide searches or finding new entities to communicate with, it will operate as a centralized system and thus the delay will follow equation 4.8. Therefore, the proposed solution in this thesis is superior to these related systems. Because even if they operate with direct connections between entities, they still maintain a centralized component which does not scaled well when performing certain operations.

4.5.3 Comparison to Other Fully Distributed Architectures

The proposed system is fully distributed, but it has advantages over alternative fully distributed systems. In the addressing of goal 1, the proposed approach is similar to the related architectures. Even though related systems such as SOFIA and COSMOS build on much more cumbersome protocols, which have a larger overhead than the DCXP protocol. In relation to goal 2, the proposed system is superior because it can provide the adaptive views, for enabling more dynamic applications. The other distributed systems offer direct dissemination of context between entities in a similar manner to that of file sharing, but only if the destination in known beforehand. Thus, such systems would have to communicate with all the entities on the system in order to create a relevance view. This can be defined as the total signaling required to create a view and can denoted by equation 4.14. This can be compared to the proposed architecture, which would only requires a smaller amount as in equation 4.15 because it can limit the amount of entities based on context proximity. To be concrete, given the same system as before with 100 000 entities having ten context values each and fifty relevant entities for a particular application. The total signaling in related distributed systems would 200 000 transmissions and 100 000 local database lookups. Where in the proposed system, this is limited to 100 transmissions and 50 database lookups. However, it is important to note that the imposed delay of the communication do not cumulatively sum to the total delay it takes for creating a view, since the communication can be performed concurrently among all entities.

$$S_{total} = \sum_{n=0}^{total\ entities} n * (S_{transmission} + S_{local\ database} + S_{transmission}) \quad (4.14)$$

$$S_{total} = \sum_{n=0}^{related\ entities} n * (S_{transmission} + S_{local\ database} + S_{transmission}) \quad (4.15)$$

The related systems have the same problem of addressing the reach of information as stated in goal 3, because there is no centralized authority. However they do not offer a distributed approach to create anonymity and control of the reach in a manageable way, which the proposed system has. Furthermore, the SOFIA and COSMOS middleware are distributed architectures which are very similar to the proposed system, which also addresses similar problems. COSMOS only focus on policy management, but the SOFIA architecture utilizes ontology based knowledge to produce relevant context views. The ontology approach has extended functionality over the context proximity calculations, used in the proposed system of this thesis. But the ontology also has a longer evaluation time to produce each view, which might not return answers within the limited time bounds of mobile applications.

4.5.4 Comparison to Networked Architectures

Networked approaches which build on integrating the context exchange into the network infrastructure can also address the goals. The CCNx project is a typical network based system, which aims to alter the router infrastructure on the Internet to enable content-based addressing. However, the proposed system in this thesis performs better, because networked approaches will require massive storage and computational power, in order to replicate and manage the global context information required for goal 1. The required computational power is especially problematic in content centric networks for a massive amount of continuously changing context and global spread of the context information. The networked approaches are also unable to create the adaptive views of goal 2, because that would entail having access to complete knowledge of the whole system in the closest router. This would mean that each individual router would have to contain the equivalent amount of data as that of the centralized storage, explained in equation 4.6. The reach of data in goal 3 is also impossible to manage, because the information is replicated throughout the whole Internet.

4.6 Chapter Summary

In summary, this chapter has presented a method for providing adaptive views of continuously changing context information to mobile application. The method has combined the dissemination method explained in previous chapter with the relevance based context scoping. Furthermore, this approach was validated in mobile applications that could utilize dynamic and continuously changing context information in a scalable manner. This approach therefore addresses goal 2 since the method can provide adaptive views of the constantly changing context information to mobile applications. In detail, these agents acquired context from local sensors, gathered context from remote sources, and traversed the network to find new context based on relevant context proximity. Furthermore, the evolving context was provided to application trough an application interface, thus it could provide adaptive views of the continuously evolving context schemas in manageable way. This is superior to related systems, because they do not have good support for neither dynamically evolving context nor provisioning of context based on relevance. This approach also addresses goal 3 because the system can in combination with DCXP manage relevance and reach on a per entity basis. This means that the end users can manage the reach of their context information in a manageable way. This is superior in comparison to relates systems, which employ different forms of authorities to achieve the same control and management.

Chapter 5

Conclusions

The recent advances in mobile technology and the inclusion of a large amount of sensors in mobile devices, has enabled pervasive applications capable of providing different services based on the status of users and entities to be achieved. However, such applications require continuous access to context information as adaptive views for intelligent decision making. Therefore, the problem faced in this thesis was with regards to the enabling of dynamic context information in mobile applications, which can both utilize context from global sensors and negotiate the context information between users and entities. Related approaches were studied and put into four different categories based on where the actual context information was stored. To be specific, they used either centralized, semi distributed, fully distributed, or networked storage. But the related systems were discarded as viable options, because they did not support the functionality in regards to scalability and adaptive views which was demanded by the proposed scenario. An alternative approach was therefore explored, which included direct exchange of context information between entities and dynamic creation of adaptive views. Furthermore, the approach was validated both qualitatively and quantitatively with proof of concept implementations and calculations on the achieved scalability. The proposed system was also compared to related systems, to further validate the approach.

This thesis explored two specific challenges, the supporting of global context dissemination and the enabling of dynamic context awareness as adaptive views in mobile applications. In detail, these two challenges was presented and addressed in their respective chapters of this thesis. The first challenge of supporting global context dissemination was explored from a peer-to-peer perspective and resulted in seven papers. These papers included the operation of the DCXP protocol, the development of the MediaSense framework, proof-of-concept applications, and evaluations of their components. The second challenge of managing dynamic context awareness as adaptive views came from the observation that there is too much changes in context for traditional systems to be feasible. This challenge was explored in three papers, which resulted in an application interface that provides adaptive views of the evolving context and an agent based architecture for performing the evolution of the context information. In conclusion, by means of these ten papers the concrete goals of this thesis have been achieved.

Goal 1 was met by means of the dissemination and negotiation method used in the MediaSense framework and in particular the context exchange of the DCXP network. This approach enables applications to utilize context information derived from global sensors, in a fully distributed manner. This approach was also validated in multiple proof-of-concept applications which utilized the system. The proposed approach was also compared to related approaches by calculating delays and database sizes, which showed that the proposed system scales better. This goal has thus contributed an approach in order to enable the dissemination of global context in ubiquitous scenarios.

Goal 2 was met by means of the agent based approach that has the ability to create continuously evolving context information. In detail, these agents could acquire context from local sensors, gather context from remote sources, and traverse the network to find new context based on relevant context proximity. Furthermore, the evolving context was provided to the application through an application interface, capable of providing adaptive views of the continuously evolving context schemas in a manageable way. This goal was studied in related approaches, which shows limited functionality and poor scalability when providing adaptive views.

Goal 3 was met by means of the combination of the DCXP network and the agent based network, in order to create context evolution based on relevance. Because each entity manages its own context locally and decides whom to share it with, it is possible to maintain control of the information flow. Furthermore, the anonymity enabled DCXP extension also provide DCXP with the means of handling privacy and reach of the information in a manageable way. Therefore, this system can both handle the reach of personal context information in a manageable way and control the flow of personal information.

From these goals, it can be concluded that is is possible to derive context information from global sensors, and to disseminate it between users and entities. It is also possible to conclude that the proposed distributed approach is superior with regards to scalability and extensibility, which will be required by next generation context-aware systems. Furthermore, this thesis can conclude that the next generation services will require a new form of access to information in applications. Such applications will require specialized access interfaces for context, because the dynamics of context information will be unmanageable with the currently available systems. By combining the three goals and the two challenges posed in this thesis, a solution to the problem of creating context awareness in context-aware applications has been determined. In detail, the solution to this problem is a distributed system which scales well and enables there to be evolving context information and adaptive context-aware applications. It supports continuously changing context and can provide access to the context information as adaptive views, which does not endanger the dynamics of the evolving context. The system is also sufficiently lightweight to run on ubiquitous mobile devices with unstable connectivity and limited resources. The proposed system's superiority compared to related systems was proven both qualitatively by identifying new features that was enabled and quantitatively by calculating different scalability metrics of the system.

5.1 Future Work

The outlook for mobile context-aware applications is good, even if the majority of commercial applications will utilize centralized systems. However, more context-aware applications will emerge and demand an open distributed access to context information. Current distributed approaches will however require additional research into security and privacy. These approaches also require extensive simulations in order to determine the stability and performance which will be involved as the number of users increases in magnitude to billions of users. Such simulations should provide an insight into the choice of distribution method, including the DHT variant and protocol choice. Further studies and development of the application interface are also required, in order to provide verification by both simulations and real world prototyping. Other approaches can also be explored, to discover additional programming patterns which are suitable for mobile context-aware application development with evolving context. However, the future for the proposed system presented in this thesis can be to enable open context-awareness in smart objects. For example, enabling autonomous agents gathering context about relevant entities, thus enabling context-aware behavior and self learning of all kinds smart objects in an entity's context proximity.

5.2 Concluding Remarks

This thesis has explored distributed approaches in order to enable context-aware applications in mobile environments. Current systems can manage context information and provide it to applications, but problems do occur with adaptive applications and continuously evolving context information. With the current explosion in the mobile market and the enormous number of in-built sensors in the present day appliances, there is greater awareness of the issues associated with billions of connected things. The mobile context-aware applications are therefore the next logical step towards truly pervasive applications with ubiquitous sensing for the Internet of Things.

References

[1] T. Kanter, S. Pettersson, S. Forsström, V. Kardeby, and P. Österberg, "Ubiquitous Mobile Awareness from Sensor Networks," in *Mobile Wireless Middleware, Operating Systems, and Applications - Workshops*, 2009, pp. 147–150.

[2] T. Kanter, P. Österberg, J. Walters, V. Kardeby, S. Forsström, and S. Pettersson, "The MediaSense Framework," in *Proceedings of Fourth Int. Conf. Digital Telecommunications*, 2009, pp. 144–147.

[3] T. Kanter, S. Forsström, V. Kardeby, J. Walters, P. Österberg, and S. Pettersson, *Ubiquitous Mobile Awareness from Sensor Networks*, 2010, ch. 16, pp. 449–463.

[4] T. Kanter, S. Pettersson, S. Forsström, V. Kardeby, R. Norling, J. Walters, and P. Österberg, "Distributed Context Support for Ubiquitous Mobile Awareness Services," in *Proceedings of Fourth International Conference on Communications and Networking in China*, 2009.

[5] S. Forsström, V. Kardeby, J. Walters, R. Norling, and T. Kanter, "Dissemination of Anonymised Context Information by Extending the DCXP Framework," in *Mobile Networks and Management: First International Conference*, 2010, pp. 57–66.

[6] V. Kardeby, S. Forsström, J. Walters, P. Österberg, and T. Kanter, "The Updated MediaSense Framework," in *Proceedings of Fifth International Conference on Digital Telecommunications*, 2010, pp. 48–51.

[7] S. Forsström, V. Kardeby, J. Walters, and T. Kanter, "Location-Based Ubiquitous Context Exchange in Mobile Environments," in *Mobile Networks and Management: Second International Conference*, 2010.

[8] T. Kanter, V. Kardeby, S. Forsström, and J. Walters, "Scenarios, Research Issues, and Architecture for Ubiquitous Sensing," in *Mobile Networks and Management: Second International Conference*, 2010.

[9] S. Forsström and T. Kanter, "Continuously Evolving Context Information by Utilizing Sensors in Mobile Environments," 2011, submitted to 1st International Workshop on Opportunistic Sensing and Processing in Mobile Wireless Sensor and Cellular networks.

[10] S. Forsström and T. Kanter, "Enabling Adaptive Context Views for Mobile Applications utilizing Dynamic Sensor Information," 2011, submitted to 4th International ICST Conference on Mobile Wireless Middleware, Operating Systems, and Applications.

[11] P. Brown, J. Bovey, and X. Chen, "Context-Aware Applications: from The Laboratory to The Marketplace," *IEEE Personal Communications*, vol. 4, pp. 58–64, 1997.

[12] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," in *Mobile Computing Systems and Applications, 1994. WMCSA'08. First Workshop on*, 1994, pp. 85–90.

[13] J. Pascoe, "Adding Generic Contextual Capabilities to Wearable Computers," in *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, 1998, pp. 92–99.

[14] A. Dey and G. Abowd, "Towards a Better Understanding of Context and Context-Awareness," in *CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness*, 2000, pp. 304–307.

[15] J. Hong, E. Suh, and S. Kim, "Context-Aware Systems: A Literature Review and Classification," *Expert Systems with Applications*, vol. 36, pp. 8509–8522, 2009.

[16] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," in *Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.

[17] D. McGuinness, F. Van Harmelen *et al.*, "OWL Web Ontology Language Overview," W3C, Recommendation, 2004.

[18] T. Strang, C. Linnhoff-Popien, and K. Frank, "CoOL: A Context Ontology Language to Enable Contextual Interoperability," in *Distributed Applications and Interoperable Systems*, 2003, pp. 236–247.

[19] K. Cheverst, K. Mitchell, and N. Davies, "Design of An Object Model for A Context Sensitive Tourist Guide," *Computers & Graphics*, vol. 23, no. 6, pp. 883–891, 1999.

[20] B. Bouzy and T. Cazenave, "Using The Object Oriented Paradigm to Model Context in Computer Go," in *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context*, 1997.

[21] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[22] B. Schilit and M. Theimer, "Disseminating Active Map Information to Mobile Hosts," *IEEE network*, vol. 8, no. 5, pp. 22–32, 1994.

[23] R. Hull, P. Neaves, and J. Bedford-Roberts, "Towards Situated Computing," in *First International Symposium on Wearable Computers*, 1997, pp. 146 –153.

[24] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: A Mobile Context-Aware Tour Guide," *Wireless Networks*, vol. 3, no. 5, pp. 421–433, 1997.

[25] L. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl5, and H. Gellersen, "Smart-Its Friends: A Technique for Users to Easily Establish Connections Between Smart Artefacts," in *Ubiquitous Computing*, 2001, pp. 116–122.

[26] S. Antifakos, B. Schiele, and L. Holmquist, "Grouping Mechanisms for Smart Objects Based on Implicit interaction and Context Proximity," in *Adjunct Proceedings of International Conference on Ubiquitous Computing*, 2003.

[27] 3GPP. (2009, December) TS 23 228: IP Multimedia Subsystem (IMS); Stage 2 (Release 9). [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/23228.htm

[28] J. Rosenberg, "SIMPLE made Simple: An Overview of the IETF Specifications for Instant Messaging and Presence using the Session Initiation Protocol (SIP)," IETF, IETF Draft, 2008.

[29] M. E. Barachi, A. Kadiwal, R. Glitho, F. Khendek, and R. Dssouli, "A Presence-Based Architecture for the Integration of the Sensing Capabilities of Wireless Sensor Networks in the IP Multimedia Subsystem," in *Proceedings of IEEE Wireless Communications and Networking Conference*, 2008, pp. 3116 – 3121.

[30] A. Kansal, S. Nath, J. Liu, and F. Zhao, "Senseweb: An Infrastructure for Shared Sensing," *IEEE MultiMedia*, vol. 14, no. 4, pp. 8–13, 2007.

[31] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," IETF, RFC 3920, 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3920.txt

[32] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," IETF, RFC 3261, 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[33] D. Raz, A. Juhola, J. Serrat-Fernandez, and A. Galis, *Fast and Efficient Context-Aware Services*, D. Hutchison, Ed. John Wiley & Sons Ltd, 2006.

[34] M. Klemettinen, *Enabling Technologies for Mobile Services: The MobiLife Book*. John Wiley & Sons Ltd, 2007.

[35] M. Presser, P. Barnaghi, M. Eurich, and C. Villalonga, "The SENSEI Project: Integrating The Physical World with The Digital World of The Network of The Future," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 1–4, 2009.

[36] H. Gellersen, A. Schmidt, and M. Beigl, "Adding Some Smartness to Devices and Everyday Things," in *Third IEEE Workshop on Mobile Computing Systems and Applications*, 2000, pp. 3–10.

[37] A. Schmidt, M. Beigl, and H. Gellersen, "There Is More to Context Than Location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, 1999.

[38] H. Gellersen, A. Schmidt, and M. Beigl, "Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts," *Mobile Networks and Applications*, vol. 7, no. 5, pp. 341–351, 2002.

[39] M. Khedr and A. Karmouch, "Exploiting SIP and Agents for Smart Context Level Agreements," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2003.

[40] M. Khedr, A. Karmouch, R. Liscano, and T. Gray, "Agent-Based Context-Aware Ad hoc Communication," in *Mobile Agents for Telecommunication Applications*, 2002, pp. 105–118.

[41] A. Toninelli, S. Pantsar-Syväniemi, P. Bellavista, and E. Ovaska, "Supporting Context Awareness in Smart Environments: A Scalable Approach to Information Interoperability," in *Proceedings of the International Workshop on Middleware for Pervasive Mobile and Embedded Computing*, 2009.

[42] P. Bellavista, R. Montanari, and D. Tibaldi, "Cosmos: A Context-Centric Access Control Middleware for Mobile Environments," in *Mobile Agents for Telecommunication Applications*, 2003, pp. 77–88.

[43] S. Krco, D. Cleary, and D. Parker, "P2P Mobile Sensor Networks," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005.

[44] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, 2009.

[45] L. Chen, C. Yu, C. Tseng, H. Chu, and C. Chou, "A Content-Centric Framework for Effective Data Dissemination in Opportunistic Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, pp. 761–772, 2008.

[46] G. Hazel and A. Norberg, "Extension for Peers to Send Metadata Files," BitTorrent.org, BEP 9, 2008.

[47] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A Knowledge Plane for The Internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 3–10.

[48] A. Karmouch, A. Galis, R. Giaffreda, T. Kanter, A. Jonsson, A. Karlsson, R. Glitho, M. Smirnov, M. Kleis, C. Reichert *et al.*, "Contextware Research Challenges in Ambient Networks," in *Mobility Aware Technologies and Applications*, 2004, pp. 62–77.

[49] C. Dannewitz, K. Pentikousis, R. Rembarz, É. Renault, O. Strandberg, and J. Ubillos, "Scenarios and Research Issues for A Network of Information," in *4th International Mobile Multimedia Communications Conference*, 2008.

Biography

Stefan Forsström was born on the 11th of May 1984 in Sundsvall, Sweden. He received his Master of Science and Engineering from the Mid Sweden University in August 2009 and his master's thesis won the Sundsvall 42 scholarship for excellent thesis work. Directly after his master's thesis, Forsström started as a PhD student in the MediaSense research project at Mid Sweden University. Where he focus on mobile application development, in particular sensor based context-aware applications.