Ripples Across
The Internet of Things

Context Metrics as Vehicles for
Relational Self-Organization

Jamie Walters

Department of Information Technology and Media
Mid Sweden University
Licentiate Thesis No. 59
Sundsvall, Sweden
2011
Akademisk avhandling som med tillstånd av Mittuniversitetet framlägges till offentlig granskning för avläggande av teknologie licentiatexamen tisdagen den 26 april 2011 i L111, Mittuniversitetet, Holmgatan 10, Sundsvall.

©Jamie Walters, March 2011

Tryck: Tryckeriet Mittuniversitetet
For Mummy

Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.

- Sir Winston Churchill, Nov 10, 1942
Abstract

The current paradigm shift in computing has placed mobile computation at the centre of focus. Users are now even more connected; demanding *everything everywhere* services. These services, such as social networking and media, benefit from the availability of context information seamlessly gathered and shared; providing customized and user-centric experiences. The distribution of context information no longer conforms to the paradigms of the existing Internet with regards to heterogeneity, connectivity and availability. This mandates new approaches towards its organization and provisioning in support of dependent applications and services.

In response to these developments, the work summarized in this thesis addresses the fundamental problem of presenting context information in organized models as relevant subsets of global information. In approaching this problem, I introduced a distributed collection of context objects that can be arranged into simple relevant subsets called context schemata and presented to applications and services in supporting the realization of context based user experiences. Acknowledging the dynamic behaviour inherent of the real world interactions, I introduced an algorithm for measuring the proximities and similarities among these context objects, providing a metric through which to achieve organization. Additionally, I provided a means of ranking heterogeneous and distributed sensors in response to real time interaction between users and their digital ecosystem. Ranking provides an additional metric with which to achieve organization or identifying important and reputable information sources. The work I present here, additionally details my approach to realizing this complete behaviour on a distributed overlay, exploiting its properties for distribution, persistence and messaging. The overlay is also utilized for the provisioning of the supporting context information.

Improvements in the ability to discover and attach new context information sources is fundamental to the ability to continually maintain expressions of context, derived from heterogeneous and disparate sources. By being able to create relevant subsets of organized data related to the requirements of applications and services in an end-point, infrastructures are realized for connecting and supporting the increasingly large numbers of users and their sources of information. Coupled with the distribution, these infrastructures realize improvements with regards to the effort required to achieve the same results. The culmination of the work presented in this thesis is an effort to enable seamless context-centric solutions on a future *Internet of Things* and thus constituting an adequate solution to the challenges raised above.
Acknowledgements

Writing and compiling a thesis is a relatively demanding task. The research that precedes it, however, provides for some of the most interesting periods in an academic’s career. Undertaking this task at Mittuniversitetet, has been for me an even more challenging opportunity. One that required me to move to a new country and assimilate to a varying and often times contrasting culture. The experience, however, has afforded me the opportunity to see and explore the application of Computer Science from a new perspective and with new expectations. A chance to create thought provoking juxtapositions in predicting and researching ideas that are expected to contribute to the evolving technological landscape.

Professor Theo Kanter has been fundamental in providing the opportunity to undertake this research and provide the support required to meet the requirements for this publication; the voice of reason and direction in the dynamic and fast paced research environment. I acknowledge this along with the contribution of Dr. Stefan Pettersson, my second supervisor, towards the completion of this important milestone.

The MediaSense research group has afforded me the opportunity to accomplish the research outcomes outlined in this document. Working with this project expanded the scope of thinking with respects to the application and usage of pervasive and ubiquitous computing devices. To, as a future scientist, grasp the expectations of academia and industry and to be in a position to stimulate thoughts and ideas that seek to enable expectations and address some of the missing blocks in contemporary research. I would like to acknowledge the contribution and support of Dr. Ulf Jennehag, Dr Patrik Österberg, Stefan Forström, Victor Kardeby and Roger Norling while being involved in the MediaSense project. Felix Dobslaw, for initiating some of the fundamental approaches, extended in this thesis. The entire department at Mittuniversitetet has been tremendous in enabling a warm environment, and encouraging my Swedish language adoption for which I am eternally grateful.

To my family, thank you so much. Mum you are an amazing woman and I owe you all that I have. When I left home to do this, I knew you were concerned, but have always understood and never discouraging. To my Nan, you are my rock. Auntie Ann, thank you, your positive energy is captivating. Helena and family, thank you. I never thought I would have found a family as warm and welcoming as mine. Pinkey and Tan: muff respec’

cf yuh wa’an gud, yuh nose haffi run

vii
# Table of Contents

**Abstract** v

**Acknowledgements** vii

**List of Papers** xiii

**Terminology** xix

## 1 Introduction

1.1 Context Awareness in Perspective ............................. 1
1.2 Problem Statement ............................................ 4
1.3 Objectives and Scope ......................................... 4
1.4 Concrete and Verifiable Goals ................................. 5
1.5 Methodology .................................................... 5
1.6 Contributions .................................................. 6
1.7 Outline ......................................................... 9

## 2 Background

2.1 The Internet of Things ......................................... 11
2.2 Pervasive and Ubiquitous Computing ......................... 11
2.3 Wireless Sensor Actuator Networks .......................... 13
   2.3.1 Sensors .................................................. 13
   2.3.2 Actuators ................................................ 13
2.4 Publish Subscribe Systems .................................... 14
2.5 Presence ....................................................... 14
2.6 Context Information ........................................... 14
2.7 Context Modelling .................................................. 15
  2.7.1 Key-Value Models ........................................ 15
  2.7.2 Mark-up Scheme Models ................................... 15
  2.7.3 Graphical Models ........................................... 15
  2.7.4 Object Oriented Models .................................... 16
  2.7.5 Logic Based Models ........................................ 16
  2.7.6 Ontology Based Models .................................... 16
2.8 Context Networks .................................................. 17
2.9 Context Proximity .................................................. 17
2.10 Resource Ranking .................................................. 17
2.11 Summary ........................................................... 18

3. Theory and Related Work ............................................ 19
  3.1 Architectures for the Provisioning of Sensor Information .... 19
    3.1.1 Scenario .................................................. 20
    3.1.2 Analysis .................................................. 20
  3.2 Distributed Models Supporting Context Awareness .......... 23
  3.3 Estimating Context Proximity .................................. 25
  3.4 Estimating Sensor Ranking ..................................... 28
  3.5 Summary ........................................................ 30

4. Approach .............................................................. 31

5. An Architecture for the Provisioning of Context Information ... 33
  5.1 The Overlay ..................................................... 33
  5.2 The Distributed Context eXchange Protocol .................. 35
  5.3 Context User Agent - CUA ..................................... 36
  5.4 Universal Context Identifier - UCI ........................... 36
  5.5 Context Storage - CS .......................................... 36
  5.6 Persistence ...................................................... 38
  5.7 Resource-Constraint Devices ................................... 39
  5.8 Evaluation ...................................................... 39
  5.9 Summary ........................................................ 40

6. Distributed Models Supporting Context Services .................. 41
# Table of Contents

6.1 The Context Information Integration Model ...................................... 41
6.2 The Distributed Context Information Integration Model ..................... 43
6.3 Context Schemata .............................................................. 45
6.4 Evaluation ................................................................. 46
6.5 Summary ................................................................. 47

7 Estimating Context Proximity ......................................................... 49
  7.1 Overview ................................................................. 49
  7.2 Single Dimension Proximity .................................................. 51
  7.3 Multi-Dimension Proximity ................................................... 52
  7.4 Realization in Distributed Architecture ..................................... 56
  7.5 Verification .............................................................. 57
  7.6 Evaluation .............................................................. 61
  7.7 Summary .............................................................. 62

8 Estimating Sensor Ranking ............................................................ 63
  8.1 Overview ................................................................. 63
  8.2 Localized Ranking ......................................................... 64
  8.3 Time Limited Localization .................................................. 65
  8.4 Global Ranking .......................................................... 65
  8.5 Approach on a Distributed Architecture .................................... 66
  8.6 Verification .............................................................. 67
  8.7 Evaluation .............................................................. 67
  8.8 Summary .............................................................. 68

9 Conclusions and Future Work ....................................................... 69

Bibliography ................................................................. 73
List of Papers

This thesis is mainly based on the following papers, herein referred by their Roman numerals:


# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Context Awareness Supporting Infrastructure</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>The MediaSense Infrastructure</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Overview of Contributions</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>Taxonomy of Research Problems in Pervasive Computing</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Overview of a Publish Subscribe System</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Overview of Context Proximity</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Overview of the Centralization of the SenseWeb Project</td>
<td>21</td>
</tr>
<tr>
<td>5.1</td>
<td>Overview of the Context Provisioning Architecture</td>
<td>34</td>
</tr>
<tr>
<td>5.2</td>
<td>Overview of P-Grid Virtual Distributed Tree</td>
<td>34</td>
</tr>
<tr>
<td>6.1</td>
<td>Context Information Integration Model (CI)</td>
<td>42</td>
</tr>
<tr>
<td>6.2</td>
<td>Distributing the Meta Model</td>
<td>43</td>
</tr>
<tr>
<td>6.3</td>
<td>Distributed Meta Model</td>
<td>46</td>
</tr>
<tr>
<td>7.1</td>
<td>Presentity &amp; Information Point Relationships</td>
<td>50</td>
</tr>
<tr>
<td>7.2</td>
<td>Single Dimension Context Proximity Cluster</td>
<td>53</td>
</tr>
<tr>
<td>7.3</td>
<td>Determining Presentity Proximity</td>
<td>54</td>
</tr>
</tbody>
</table>
List of Tables

5.1 The Primitives of the Distributed eXchange Protocol . . . . . . . . . . 37
6.1 The Main Components of the CII Model . . . . . . . . . . . . . . . . . 43
7.1 Verifying Context Proximity - Results 1 . . . . . . . . . . . . . . . . . 58
7.2 Verifying Context Proximity - Results 2 . . . . . . . . . . . . . . . . . 58
7.3 Verifying Context Proximity - Results 3 . . . . . . . . . . . . . . . . . 59
7.4 Verifying Context Proximity - Results 4 . . . . . . . . . . . . . . . . . 59
Terminology

Abbreviations and Acronyms

XML eXtensible Markup Language
RDF Resource Description Framework
IMS IP Multimedia Subsystem
SQL Structured Query Language
3GPP Third Generation Partnership Project
IPTV Internet Packet Television
WSAN Wireless Sensor Actuator Network
WSN Wireless Sensor Network
RSS Really Simple Syndication
ATOM The Atom Syndication Format
DHT Distributed Hashtable
DNS Domain Name Service
OWL Web Ontology Language
SIP Session Initiation Protocol
SIMPLE Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions
JSON JavaScript Object Notation
UCI Universal Context Identifier
URI Universal Resource Identifier
CII Context Information Integration Model
DCII Distributed Context Information Integration Model
CUA Context User Agent
OODB Object-Oriented Database
RDB Relational Database
CS Context Service
SR Sensor Ranking
MDP Mobile Data Proxy
lCUA Limited Context User Agent
rCUA Remote Context User Agent
Chapter 1

Introduction

The increasing interest in the provisioning of applications and services that deliver experiences based on context mandates research into methodologies, architectures and support for delivering the information required. Constraints on service delivery with respect to real-time availability underpins any such solutions. This thesis investigates approaches, methodologies and solutions for enabling context dependent applications and services on a future Internet of Things.

Such a connected things infrastructure is expected to have an installed device base exceeding billions [SCFW10], and will be expected to support a wide range of context centric experiences ranging from personalized and seamless media access, to intelligent commuting or environmental monitoring. This will incorporate devices such as televisions, refrigerators, cars and public transport. All merging towards the paradigm of everywhere computing [LSK+08]: the seamlessly connected new world.

The work contained in this thesis report revolves around the creation of the distributed connected architectures required to support such services with a focus on creating new approaches to searching and browsing in context centric networks. A move towards creating such infrastructures will require an understanding of the inherent problems of finding relevant information within real time constraints in order to deliver user experiences that are current and relevant. Novel approaches are proposed with regards to estimating the relevance and importance of information sources in a vast and connected architecture.

As a part of the MediaSense research project (See Paper[II]), the contributions detailed in the thesis create a part of a wider picture, a dynamic connected infrastructure supporting users and services demanding current and reliable context information.

1.1 Context Awareness in Perspective

Anind Dey, in his paper Understanding and Using Context [Dey01], contributes significantly to the understanding of context information and its ability to aid in defining
the advances in ubiquitous and pervasive computing research. This introduced the two concrete definitions; pillars of modern context aware computing, defining Context as:

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.

and Context Awareness as:

A system is context aware if it uses context to provide relevant information and or services to a user, where relevancy depends on the user’s task.

The continual realization of these definitions in practical applications has largely been focused on the ability to derive accurate and timely representations of a user’s state of being from an intricately woven ecosystem of sensor and actuators. These are largely realized today as Wireless Sensor Actuator Networks (WSANs) and are in the future expected to blanket the globe, existing both as fixed and mobile installations [Sta08]. Enabling context awareness utilizing massively deployed sensor networks mandates the need to investigate architectures capable of enabling the required support. Figure 1.1 illustrates an overview of such a context enabling architecture, enabling the incorporation of sensing information in the delivery of applications and services.

Hong [HL01], outlined approaches to designing and realizing infrastructures that support context centric applications and services. Such infrastructures, he motivated, must be preferred to collections of ad hoc tool-kits and framework and be able to satisfy the five requirements of context aware computing, summarized as follows:
1.1 Context Awareness in Perspective

![Figure 1.2: The MediaSense Infrastructure](image_url)

1. Designing the data formats and network protocols to be simple enough that they can be implemented on virtually any platform, but also rich enough to represent the majority of sensor and context data.

2. Building the basic services in the infrastructure, including automatic path creation and proximity-based discovery.

3. Finding the middle ground between smart devices and smart infrastructures, finding the right balance of responsibilities between the two.

4. Scoping of sensor and context data to ensure security and privacy.

5. Building an infrastructure that will scale up gracefully for large numbers of sensors, services, devices, and people.

Within the constraints of these five characteristics, can be realized a wide range of applications that can be enabled including multimedia, telephony and social networking which is described by [JM09] as being the missing block. The final component in the move to realize socially connected infrastructures where explicitly generated context is complemented by implicitly attached context dimensions derived from scalable and reliable sources.

The MediaSense project explores concrete realizations of the enabling solutions demanded by such consumers by creating such an infrastructure, shown in figure 1.2 aimed at unifying existing presence systems and content existing across the internet with current context information to drive the development and adoption these services. The remaining work in this thesis contributes to addressing the first, second, third and fifth requirements, with an emphasis on the second.
1.2 Problem Statement

The Internet of Things encompasses the enabling of an *everything everywhere* society. The ubiquity of context information presents problems for its efficient acquisition, provisioning and usage towards the realization of dependent applications and services. This information needs to be organized in a manner capable of supporting the dynamic and distributed behaviour reflected by the real world interactions it defines and support. However, the organization of the data must be relevant and accurate in the end points. As a consequence, these subsets of data must be derived relevant to the currently observed states in the end points and the requirements of its local applications and services. Further, there exists the problem of being able to accomplish this in a scalable manner also capable of supporting sensors information provisioning. This is required to liberate approaches from non-scalable properties which can place a limit on the freshness and availability while avoiding a dependency on web services utilizing DNS.

1.3 Objectives and Scope

The work I present in this thesis encompasses research primarily aimed at building a *Connected Things Infrastructure*, an enabler to connecting and composing context information in real time scenarios. In addressing the problems described earlier in Section 1.2, my research was firstly concerned with the provisioning of context information from current and existing data types. It assumes that our interaction with context information will increase substantially while current means of representing sensor data will remain largely unchanged, limiting my scope to the presentity/watcher model of presence systems. The presentity/watcher model mandates a solution that considers the user as being the focal point of interaction and service delivery. The distributed provisioning of information was central and issues that may arise concerning authorization, authentication, security or privacy were not studied.

In achieving distributed organization, my work was focused on the ability to take an existing model and achieve distribution over any base architecture that existed. In doing this, I was further interested in how such a solution could interpolate with existing presence infrastructures. However, the key underlying focus was the ability to create distributed pools of objects that were searchable, browsable and composable. The use of models for reasoning and knowledge discovery was also not investigated.

Finally, of interest in this direction was the derivation of subsets of context information specific to a presentity at a given point in time. The ability to derive metrics on the context proximity between any two entities and enabling solutions that continually evaluate and evolve representations of context neighbours within application specified parameters. Context proximity was limited to and concerning this user and his subset of data. A global proximity factor was never investigated due to the underlying issues with converging values over a distributed heterogeneous architecture. I limited proximity to a personal computation where a value was only meaningful to a presentity and to a specific context. Sensor ranking however, was a global ap-
Concrete and Verifiable Goals

The resulting work did not consider issues that may arise due concerning authorization, authentication, security or privacy. This research delivers an overall approach to a distributed context-centric architecture that can enable applications and services to deliver real-time solutions in response to a user’s interaction and state of being, both with respects to the environment and to other desirable users.

1.4 Concrete and Verifiable Goals

The underpinning research is expected to offer, to end points, a manageable collection of context information as a relevant subset of the global context information. In accomplishing this, the following must also be met:

- Construct a distributed approach to modelling and sharing context information, presenting this in a scalable manner permitting evolution reflecting the interactions of the underpinning entities.
- Introduce and verify an algorithm for estimating context proximity over multiple dimensions, addressing its applicability in a distributed architecture. Such an algorithm should offer improvements in computational operations when compared to alternative approaches.
- Introduce and verify an algorithm for ranking sensors along with its applicability in a distributed architecture. Such an algorithm should offer improvements in computational operations when compared to alternative approaches.

1.5 Methodology

In approaching the problems in Section 1.2 I began at the basics. The first task was to identify an optimal solution for the provisioning of context information. The existing DCXP architecture, achieved acceptable response times, however at the expense of an overlay with a complexity of $O(N)$. An investigation into a substitute overlay was conducted as well as support for incorporating a distributed object model directly into the architecture exploiting any benefits gained by the new overlay.

I investigated existing solutions to finding relevant information on the Internet. However with none suitable to the application of context with respects to its heterogeneity and fluidity, I set about creating an algorithm for finding similar entities in real time while remaining cheap with respects to the underlying operational costs. This was verified and compared against existing generic distance functions.

Acknowledging that a present entity will still observe a largely homogeneous distribution of sensors with no regards to their importance or reputation, I set out to create
a distributed algorithm for ranking sensors based on their usage. Noticing the similarities between this problem and document ranking, I examined relevant approaches selecting a simple document frequency inspired approach as opposed to the more commonly used Page Rank inspired approaches as the dynamic behaviour of context information renders these approaches non-scalable. This was verified with respects to operational costs and its ability to compute a reliable value.

1.6 Contributions

The main contributions of this thesis to the research community is summarized as a list of publications as follows and illustrated in figure 1.3. This is represented as a flow chart with arrows indicating how each publication contributed to the subsequent publications and the final outcome of the thesis.

Publication I

Distributed Context Support for Ubiquitous Mobile Awareness Services

Introduced the Distributed Context eXchange Protocol (DCXP) as the underpinning architecture of the MediaSense approach. DCXP is positioned as a suitable alternative to the existing centralized solutions enabling context-aware computing. The contribution addresses issues of scalability and the ability to derive and disseminate context information while maintaining real-time constraints. The solution along with its principle operation is demonstrated and evaluated in a sample ubiquitous mobile awareness service. In this paper, I contributed the distributed aggregation of sensor information through the use of random peers implementing Dataminers, Database Agents and Database Clients as a move towards a fully distributed data persistence solution.

Publication II

The MediaSense Framework

Examines the trends in the rapidly evolving mobile telecommunication environment with regards to everything, everywhere computing. It positions the MediaSense framework as a feasible solution towards the problem of provisioning the intelligent delivery of any information to any host, anywhere. This based on context-aware information regarding personal preferences, presence information, and sensor values. It further brings into focus research issues with regards to achieving seamless delivery, primarily of multimedia content, and multi-modal services via heterogeneous connections. The solution along with its principle operation is demonstrated and evaluated in a sample ubiquitous mobile awareness service. In this paper, I author contributed the distributed aggregation of sensor information through the use of random peers implementing Dataminers, Database Agents and Database Clients.
Publication III

Dissemination of Anonymised Context Information by Extending the DCXP Framework

Proposed a solution for introducing anonymity within the DCXP architecture as a means of enabling context dissemination where issues of privacy might arise in a distributed solution. The solution sought to maintain the real-time properties achieved while minimizing any impact on the ability of DCXP to scale well. In this paper, I contributed the end point anonymity solution through the use of a token ring approach to adding and removing messages, in an effort to conceal end point originators and recipients.

Publication IV

The Updated MediaSense Framework

Provides an update on the MediaSense project. The focus of this paper is the overall framework; the handling of user profiles; the management of context information; real-time distribution and proof of concepts. In this paper I contributed the distributed composable context schemas and user preference profiles as groundwork towards enabling the extraction of meaningful metrics for connecting and composing context solutions.

Publication V

An Object-Oriented Model in Support of Ubiquitous Mobile Awareness Services

Addresses the issues with regards to creating intelligent context-aware services within the DCXP architecture. It presents a shared object-oriented meta model for a persistent agent environment. This model seeks to facilitate the creation of ambient intelligence underpinned by context-aware agents whose actions are based upon context and social information complemented by the behaviour of the agents themselves. In this paper I contributed towards a refined approach to the model and its realization on the DCXP framework.

Publication VI

Location-Based Ubiquitous Context Exchange in Mobile Environments

Introduces a lightweight model for composing and maintaining unstructured location-scoped networks of peer-to-peer nodes, which gossips in order to ensure quality of service for each user. The model is implemented in a prototype application running in a mobile environment, which is evaluated with respect to real-time properties. I
contributed to the general algorithm for achieving such an ad hoc gossiping approach by creating personal groups and utilizing gossiping for discovering and maintenance.

**Publication VII**

**Scenarios, Research Issues, and Architecture for Ubiquitous Sensing**

Proposes an information-centric architecture, building on the previous work with distributed object models. This is proposed as solution towards real-time ubiquitous sensing, capitalizing on the proposed locator/identifier split, thus extending a previously proposed Network of Information (NetInf) approach. From this we identify the challenges for which we present work-in-progress within the framework of the Medi-aSense project. I contributed the distributed approaches towards the object/value split implementation.

**Publication VIII**

**Distributed Context Models in Support of Ubiquitous Mobile Awareness Services**

Details an extension of such a model into a distributed architecture co-located with context user agents. This arrangement provides clients with a model schema which is continually evolving over sensor domains. In addition it proposes the use of so called context schema to represent an accurate temporal view of a user’s context with
respect to the available sensors and actuators. I contributed the main content of this paper including the distribution of the context models, interoperability and further changes to the existing DCXP approach needed to realize this.

Publication IX

Evolving Presentity-Based Context Schemas by Estimating Context Proximity

Introduces an algorithm for estimating the context proximity among presentities, which optimizes the ability to navigate and create schemata of entities relevant to, and expressing the current context of a presentity. This paper additionally proposes how an extension of the distributed gossiping algorithm can be used enable complete schemata of entities as one traverses the vast and dynamically connected things infrastructure. I contributed author contributed most of the content in this paper, including scenarios, algorithms and distribution logic.

Publication X

The MediaSense Framework: Ranking Sensors in a Distributed Architecture

This paper presents an approach to calculating sensor ranking based on general usage patterns of users in a context-centric architecture. I contributed most of the content in this paper, including scenarios, algorithms and distribution logic.

1.7 Outline

The thesis is organized as follows: Chapters 2 and 3 addresses some of background information and the related work in the research area along with some of the author’s contribution. Chapters 4 through 8 to details the author’s main work in the research area and summarizes the contribution detailed in the attached publications. Finally, Chapter 9 summarizes the conclusions arising from the work undertaken in this thesis and raises the possible avenues for future work.
Chapter 2

Background

Creating infrastructures for enabling context dependent services relies on a common understanding and appreciation of the enabling technologies. This chapter looks at some of the supporting technologies required in order to realize any systems or set of tools towards context-aware computing. It firstly addresses the general areas of the subject area, mentioning key research issues and advancements with an outlook on future expectations.

2.1 The Internet of Things

The Internet of Things [SGFW10], is the realization of a connected things infrastructure in future Internet landscape. Such a realization is underpinned by an expansive network consisting of billions of connected things; where a thing might be physical or virtual and represents some source information existing in the real world or derived and represented on the Internet.

An Internet of Things would be supported by open, interoperable protocols and components; intelligent and expressive interfaces along with a responsive real-time interaction. All contributing to the creation of application and service solutions capable of self configuration and dynamic context-centric compositions. From this, users of such an Internet could demand and expect, personalized interaction with their digital ecosystems and the real world. This enabling a truly seamlessly connected society.

2.2 Pervasive and Ubiquitous Computing

Pervasive computing encapsulates the branch of computing concerned with the concept of everywhere computing. First coined by Mark Weiser [GBSB99] in 1988, the area of interest is trending towards a paradigm where computing is emancipated from the confines of traditional desktop and server paradigms to being small, present but
still intelligent. Pervasive computing devices often exist as small, sometimes undetectable computers even at the nano scale. These devices are usually embedded into other everyday devices such as television, mobile phones, appliances or even the infrastructure and the people themselves.

Satyanarayanan [Sat01] examined the future challenges of pervasive computing which is summarized as a taxonomy in Figure 2.1. This early work painted an overall picture of the directions of research that are required in order to realize a globally accepted and functioning solution towards pervasive computing. This thesis addresses some of the factors contributing to Distributed Systems and Computing with an emphasis on:

- Mobile Networking
- Mobile Information Access
- Adaptive Applications
- Location(Context) Sensitivity

The work contained in this thesis however, deviate from the concept of Location Sensitivity to the general concept of Context Sensitivity, highlighting the contributions of [SBG99] in broadening the approach to deriving context outside of the confines of spatial location and positioning solutions. Such an ability to derive representations of context has been closely connected with the advances in Wireless Sensor Network technologies.
2.3 Wireless Sensor Actuator Networks

A Wireless Sensor Actuator Network (WSAN) refers to a collection of networked sensors and actuators acting together to monitor and maintain physical environmental conditions. These can include, temperature, humidity and pressure. Typical deployments of wireless sensor networks range from vehicular and traffic management applications in urban landscapes [CK03] to forestry and wildlife monitoring scenarios [SOP+04]. With regards to this thesis, more applicable deployments include home, office and health environments for monitoring and maintenance purposes [Aky02].

WSANs are a vital key to...advancing developers and users alike toward truly pervasive computing [PK00]; providing a means by which to deduce the current environmental conditions or affecting an entity of interest and a complementary mechanism with which to effect changes within the immediate environment of an entity in response to this information. While not always required as a complementing pair, sensor and actuators in such networks are essentially the applications’ window to the world. Advances in this area include sensor fusion [BIB97], permitting more accurate context representations as well as network resilience [GGSE01], security [CKM00] and privacy [CP03]. All factors contributing to cementing the importance of WSANs within the pervasive computing paradigm.

2.3.1 Sensors

Sensors; the small devices at the heart of context-centric computing. These are often very simple devices capable of measuring a tangible environmental variable such as temperature and location; relaying this information as a digital record. Essential components of wireless sensor networks, sensors are becoming increasingly smaller and less reliable on centralized power and resources. Sensors, today have achieved some level of pervasiveness; located in mobile phones, cars, purpose built infrastructures or even randomly distributed in the wild. A future connected society is expected to have even smaller sensors [KFZ+00] numbering into billions, embedded into buildings and implanted in the general populace [WLLP02, WD08]. From the perspective of an application, a sensor may also be virtual [DC05]: collating values from other physical sensors. Sensors are usually complemented by actuators.

2.3.2 Actuators

Actuators are equally small devices at the heart of context-centric computing. Unlike sensors, actuators permit the modification of environmental conditions or aid in delivering a required service to a user. Common actuators include, thermostats or light switches, audio sources or even implanted devices controlling heart rate or glucose levels. Together with sensors they complete the loop required for detecting and effecting changes in an environment in response to some preferences. Like sensors, actuators are expected to becoming increasingly more pervasive in a future connected things society; embedded into buildings, people or the natural environment.
2.4 Publish Subscribe Systems

The work contained in this thesis is to a large extent underpinned by the principles of publish/subscribe systems. Such systems provide a means of disseminating information where there usually exists no a priori knowledge with regards to information requirements [HGM04]. On the internet, typical examples include RSS or ATOM feeds used to disseminate news and event information. However, publish/subscribe systems are seen as key to advancing context information distribution in mobile ad hoc networks [FR07]. At the base of this solution as shown in Figure 2.2, an information source enables an interface for sinks to make a subscription request; if approved, the source periodically sends updated information to the sink based on the subscription request. The source owner has no requirement to know this in advance of the request. One disadvantage of such an approach in large infrastructures is the resources required by sources in order to fulfill large numbers of requests. However approaches such as [FR07] proposes solutions to addressing these issues in mobile networks.

2.5 Presence

Presence or presence information revolves around the ability to define the state of availability of an entity in a network. Presence information in messaging systems range from simple statuses such as Away, Online or Offline to the inclusion of more fine-grained context information such as location sensing and calendar information [PLdH03]. The general model of presence systems details the implementation of two main entities Presentities and Watchers [Chr02]. The Presentity is regarded as the source of the presence information; the entity of which a service or application requires presence information in order to deliver a service. The watcher is regarded as the entity requiring a view over the presence information of a second entity in order to provision an application and service in response to presence.

2.6 Context Information

Context Information is the general term for all information related to and describing the current state of being of an entity of interest. At its base, this is simple information concerning location, time and interaction with other entities. This can however be extended to include more contemporary representations of what constitutes context information and include messaging status and availability [SBC99]. Context information drives the move towards enabling awareness in the delivery of applications.
and services within the pervasive computing paradigm. Here, there is an emphasis on methodologies that model context information and provide means of deriving metrics permitting reasoning and deduction \cite{HIR02, HRS02}.

## 2.7 Context Modelling

Context information in its raw state provides for little useful utilization with respects to service or application provisioning. The concept of modelling context provides users of context-aware systems, new opportunities for realizing solutions that can digest context information represented in a descriptive and meaningful way. Context modelling is recognized as key in the evolution towards the enabling chain in Figure 2.1 \cite{SLP04}. Context modelling in current pervasive systems is accomplished using one of the following methodologies \cite{SLP04}:

### 2.7.1 Key-Value Models

Key-Value models are the simplest approach for modelling context information. The key-value pair is centred on delivering a value to an application, where it is represented as an environmental or application variable. These, like simple values in a database table, are relatively easy to manage, distribute and manipulate with application requirements. Common examples in a distributed approach might be key-value pairs on a distributed hash table such as Chord \cite{SMK01} or Pastry \cite{RD01}.

### 2.7.2 Mark-up Scheme Models

Mark-up scheme models are focused on constructing context models using standard mark-up languages such as the XML and RDF mark-up languages. Tags in scheme documents organize entities into hierarchies and define relationships between any two elements. Later work in this area has employed languages such as XMPP for constructing profiles and defining user/services relationships as well as recording and disseminating context and presence information. \cite{HBS02} details one example of such an approach.

### 2.7.3 Graphical Models

Creating graphical models for context representation is largely focused on the utilization of UML modelling as a means of representing the complex relationships that exists in a context-centric solution. While graphical models provide no real reasoning or computational advantages, they provide for an excellent means of assembling the entity/relationship approaches underpinning a context network. Such models can be used as the base for creating programmatic realizations of any embedded or inferred knowledge and serve as a reference point for enforcing schema rules and logic within such an architecture.
2.7.4 Object Oriented Models

Object oriented approaches to context modelling seek to build dynamic models by exploiting the existing principles of standard object oriented programming. Such an approach inherits the benefits of object re-usability and encapsulation. This permits an architecture with details hidden inside shared objects exposing interfaces and methods for interaction and usage. Further, solutions adapting this approach can describe relationships and degrees of relationships between objects, with such relationships being dynamic and even inheritable. Encapsulation also allows for high levels of abstraction between a physical sensor and its object implementation, liberating application developers from needing to interact with a wide an varying heterogeneous collection of sensors and their interfaces. Object serialization permits connected architectures to construct and deliver objects where required, concealing private information such as the physical sensor password from third party interests. Such a model enables a high level of dynamic behaviour; and when coupled with an object oriented persistence, a responsive and scalable solution.

2.7.5 Logic Based Models

Logic based modelling expresses context as a set of facts, expressions and rules. Such a model is formalized permitting inference from the facts contained within. Changes to the context information is applied as a modification of the facts represented in order to comply with the rigorous formalities of such a system.

2.7.6 Ontology Based Models

Ontology based approaches permits modelling of context information as it would be experienced within the real world. Ontologies can be represented in a multitude of ways including mark-up languages such as RDF and OWL to object oriented approaches and even graphical models. Ontologies can flexibly define the existence and complex interaction of the elements contributing context information. Degrees of interaction and relationships, hierarchies and limitations encourage the construction of models facilitating reasoning, inference and knowledge discovery, critical possibilities for any context-centric architecture.

Models representing context information, however requires that they are capable of responding within real time constraints and providing the ability to enable inference, reasoning and knowledge discovery over the domain of information. Further research [HIVI] has also looked at methods for deriving more accurate representations from imperfect or incomplete context information. The work in this thesis, however is focused on enabling models that can be distributed while maintaining real-time properties as well as exploring avenues for model interpolation in pervasive systems.
2.8 Context Networks

A context network may be summarized as the creation of networks of information built on the unification of WSANs and elements of context awareness [TDPA]. The main aim of such networks is to provide an architecture for enabling applications and services dependent on context information derivable from ubiquitously deployed WSNs.

2.9 Context Proximity

Proximity in context-awareness computing has largely been focused on spatial proximity. The concept of proximity, or degree of closeness, between any two entities underpins the ability to deliver experiences connected to presence information. The implicit approach that location is central to context proximity is cemented in the work done by [ASH03, BKr03] and [HMS+01]. There are however avenues for exploring other, more abstract definitions of what constitutes context proximity through the exploration of the multiple dimensions of context. That entities are within context proximity, while not considering spatial distance is mandated by [SBG99] which identifies with the general approach in this thesis.

2.10 Resource Ranking

Ranking resources provides improvements in the ability search, browse and locate content on the Internet [CR04]. Early document ranking approaches on the Internet existed merely as lists, sorted indices or simple document frequency based searches [SEKN92]. Later work, such as pioneered by Google [Goo10, BP98], introduced more autonomous ranking optimization by examining the links between resources as an implicit indicator of resource relevance. Such approaches now underpin most modern search and browse solutions, where a resource earns a higher, more relevant connection to a surfer, based on its connections and connections to resources favoured
by the surfer.

2.11 Summary

This chapter summarized some of the background work in the areas covered by this thesis. This provides a point from which to examine the current work in the area and the theoretical principles by which they are underpinned.
Chapter 3

Theory and Related Work

Creating infrastructures for enabling context dependent services relies on a common understanding and appreciation of the enabling technologies. This chapter examines some of the theories supporting the engineering of such systems or sets of tools towards context-aware computing. It firstly addresses the general area of provisioning sensor information as a means of driving pervasive and ubiquitous computing; summarizing key research issues and contributions with an outlook on future expectations. It then proceeds to take a closer look at a focal element of pervasive computing and of the thesis itself; context awareness.

The chapter then examines the contributing elements of context-aware computing; with a brief overview of and its importance in realizing architectures engineered towards a realization of this new context-centric paradigm. These include sensor networks, connectivity and service delivery components as well as modelling and quality of service targets. It further takes a brief look at data storage and retrieval as well as the principles of centralized, distributed and peer-to-peer computing paradigms.

Finally, towards the end of the chapter, the work in this thesis is motivated by an examination of the shortfalls of existing and trending approaches to similar research problems. By addressing the principled ideologies and scientific facts supporting the research area, I seek to create a common understanding of general and accepted approaches. This further highlights the critical shortfalls of contemporary research in the area, providing a unified set of challenges that must be addressed in order to reach a feasible solution.

3.1 Architectures for the Provisioning of Sensor Information

The provisioning of context information underpins any move towards an everything, everywhere society. Approaches towards this have been previously explored using centralized or broker based architectures. This section examines these approaches
through the use of scenario and motivates the need to investigate alternatives in realizing a global provisioning platform for the Internet of Things.

3.1.1 Scenario

Helena lives in a modern apartment in Stockholm. Being a connected city within a future connected society, Helena has at her disposal multiple sources of information capable of providing her with the knowledge required to fulfill her daily tasks. She is equipped with a broadband connection and a context agent portal in her apartment.

When Helena awakes in the morning, her tablet computer recommends a breakfast recipe based on the weather outside and something that is not on the lunch menu at her job. She turns on the radio to listen to the local radio programme while eating breakfast. On leaving home, this gets transferred to an IP based audio stream to her mobile phone. On embarking the bus, the same audio stream is available, to which she connects over the bus’ WIFI enabling her to receive announcements being broadcast intermittently about traffic conditions and delays into work.

Her work based context system, recognizes that she is being delayed due to a traffic accident and redirects mobile-capable communications to her mobile device while setting appropriate presence statuses for those which cannot be redirected. This information is updated in her calendar and forwarded to all standing meeting requests. When she arrives at work, this is detected and her communication and presence information is altered to reflect this.

3.1.2 Analysis

With such scenarios becoming more prevalent in an everything, everywhere society; necessity mandates the creation of supporting architectures. The provisioning of sensor information is seen as a basic requirement for enabling information and services that respond to changes in context of state or being. Current and existing research in the area have been proposed or built on the principles of middle-ware solutions enabling acquisition and dissemination of useful information. [HM06] summarizes the challenges facing sensor information middle-ware ranging from scalability to security, privacy, openness and ease of use. Earlier research such as Mate [LC02] were regarded as being non-flexible with regards to heterogeneity and usage while others such as TinyOS [LMP+05] and Mires [SGaV+04] were only partially compliant with regards to the challenges described by [HM06]. While approaches such as Mires, provided for publish/subscribe interfaces, the needed existed for more heterogeneous approaches to the provisioning of sensor information in direct support of context.

Early research such as the MobiLife Project [FPN+05] broke the tightly coupled connections between context information systems and wireless sensor networks. The physical sensors could now be abstracted from the data interface they expose, providing applications with a single interface to an otherwise heterogeneous collection of supporting sensors and their technologies.
Current approaches to the problem such as the SenseWeb project \cite{KNLZ07} provides for an \textit{infrastructure for shared sensing}; a portal for the aggregation and dissemination of sensor information on the internet. The SenseWeb approach, as shown in Figure \ref{fig:CentralizationOfSenseWeb}, utilizes centralized server clusters to provide a data storage and access portal realized through the use of web services with public access. This permits the creation of \textit{mashups} deriving application sensor data from a so called \textit{GeoDB}, a centralized SQL based indexing database. The centralization of SenseWeb as noted by \cite{KNLZ07} raises issues with regards to its ability to scale with the demand that would be required for a large scale provisioning system. Such scalability issues are further inherent in database implementations where there remain challenges to reducing query response times with increases in the amount of data being stored. Such challenges reduce the effectiveness of solutions such as \cite{BGS01} which attempts to solve the problem of sensor data provisioning and querying through the implementation of a database solution.

Other approaches such MobiScope \cite{AAB07} sought to federate sensor information from multiple disparate sources in order to support dependent applications. This however, afforded some improvements over previous work such as the CarTel project described in \cite{HBZ06}. The CarTel project provided no guarantees with respect to the real-time availability of sensor information. It instead relies on network caching and the opportunistic delivery of information between sensor sources and its centralized repositories. While it enabled the creation of applications based on sensor information, the resulting applications and services could not be delivered based on current and changing sensor information. Unlike CarTel networks, Mobiscopes were continually connected to the Internet keeping sensor information current. However, while the sources were distributed and heterogeneous, access to the aggregated sensor information was facilitated through the use of centralized repositories. This again raises performance and scalability issues where real time service delivery targets must be met.
Distributed approaches such as [BC10] explore the option of building context provisioning solutions using DHTs. However, while a DHT provides for a more scalable and resilient approach, it relies on deterministic hashing algorithms for achieving the distribution, indexing and locating of information. Consequently, this places a limit on their ability to utilize self-organization towards realizing a more homogeneous distribution of information located on the overlay. With regards to persisting context information, an additional disadvantage of DHTs is their inability to support queries of a range of values, critical in scenarios where John might be trying to locate a service in some approximate area or over a series of context values. While solutions such as [RHS03] have sought to address this problem, this is not done natively, mandating the implementation of additional layers of complexity on the existing overlay. DHT-based implementations such as Chord are limited to a searching complexity of $O(\log N)$ [DB04]. Solutions seeking to provision current and relevant information mandates investigations into alternatives capable of realizing improved response times.

The approaches so far fail to fulfill the requirements in Figure 2.1 with regards to distribution, high availability and scalability in a satisfactory manner. This leaves open the need to create a decentralized architecture capable of provisioning sensor information in real time.
3.2 Distributed Models Supporting Context Awareness

The need for real-time access to sensor information motivates research into supporting systems and technologies for scalable sharing on the Internet of Things in real-time. Furthermore, support of context-aware applications and services must handle models of sensor information. As there continues a realization of the Internet of Things [SGFW10], dynamic yet robust context-centric architectures must be created in direct response to the need to fulfill the demands for information created by the integral user-services relationships. The work behind this paper is further mandated by the fact that services benefit from having access to information regarding both a user’s situation and intentions, as well as information regarding the user’s social activities in order to support the user in achieving tasks.

In the realization of applications and services require the organization of this context information into models, presenting information in a manner that supports reasoning and knowledge gathering. Solutions such as Mobilife [Kle07] permits this organization. However this is realized through the use of web portals on the internet and is consequently dependent on DNS as a means of locating these service portals, users and applications. Continual issues with DNS with regards to its availability due to DoS attacks and configuration errors raises questions about its continued suitability and prompting research into Distributed Hash Table (DHT) [PMT06] alternatives. We are therefore in search of solutions that are liberated from DNS while still permitting knowledge modelling and reasoning.

Solutions realized through the distribution of a relational database and making use of the advanced research in database distribution is not applicable. This, as such a distribution assumes communication reliability in order to maintain database integrity across wide area networks which cannot be guaranteed in heterogeneous mobile scenarios [Bar99], [Ulu98]. This is also undermined by the fact that relational databases are highly inefficient for supporting real-time data manipulation, evolution and querying.

Approaches such as The Context Toolkit [SDA99], provides for so called context widgets, abstracting between the user and the implementation and logic required to access and represent context information. This can involve very complex logic outside of raw sensor information such as a notification when a certain threshold has been met or exceeded. The Context Toolkit however uses attribute/value triples for storing context information which is a relatively simplistic approach without the level of expressiveness that would be required to support applications and services acting upon the context of a user.

The SOFIA project [TPSBO09] is aimed towards the creation of smart spaces capable of being utilized by context aware applications and services. It provides a middle-ware solution based in RDF ontologies. With further filtering of information based on context, it offers the ability to deliver applications and services in response to current context. However it employs the use of information brokers which are tasked with the brokering of information from ontologies and provisioning this information to end-points. This brokering approach impacts on the freshness of the data available
to end points. It further uses relational databases for ontology persistence which subsequently subjects it to the performance issues with regards to relational databases in real time scenarios.

The SOCAM middleware [GPZ05] provides a solution for creating context-aware mobile services based on information organized into models. This however is centralized with a dependency on web service portals on the Internet and thus susceptible to the issues with DNS and scalability raised previously. The Hydrogen Project [HSP+03] realizes a middle-ware based on an object oriented ontology for the realization of context centric services. It implements a multi-layered architecture abstracting between information collection, management and usage. It is a distributed approach permitting devices to exchange context information when within close range. However, it does not make use of distribution with regards to the provisioning of sensor information and devices are constrained to using only information located locally. The limitations of the mobile devices themselves places further restrictions on processing and storage.

The approaches discussed are endowed with limitations with respects to providing models for representing context information in real-time. Such models must be expressive, scalable and dynamic without a dependency on or assumption of the availability of web service portals on the Internet.
3.3 Estimating Context Proximity

Hong [HL01] emphasized the importance of proximity-based discovery in order to permit context aware applications to discover nearby sensors or points of information. He pointed out that approaches requiring sensors to be known and connected in advance are not realistic as applications would not necessarily be aware of the context information that it would encounter at any specific time. Improvements in wireless connectivity, he argued, would permit sensors to be more liberated from infrastructural issues and be accessible, connectible and composable in an ad hoc manner.

Previous work sought to enable such derivation of proximity seamless connectivity through the incorporation of sensors, actuators and other intelligent artefacts in the real world. Research such as AmbieSense [LW05] enabled this through the use of embedded context tags. Such tags are added to artefacts such as public and private spaces, lounges, restaurants, etc. A user is expected to interact with such tags in order to derive some proximity to an artefact and perhaps by deduction, another user. This provides invaluable information for spaces not covered by the more ubiquitous GPS technologies. However the need to install and find physical tags serves to add an extra layer of complexity between users and the services they require. This relegates such an approach to a supporting or complementing technology in a general context aware solution, as opposed to a solution in its entirety.

Approaches to more ad hoc determinations of context proximity include the Smart Its project [HMS+01], [BK+03] or [FMT+02], where objects are embedded with sensors and some action, typically a shake, is performed on the objects in order to indicate them being within context proximity. These, however also require specific hardware and are unable to directly utilize the context information readily available from other ubiquitous sources. This in itself, creating a barrier to large scale user adoption.

The ability to derive context from existing sources is an integral part of projects such as Senseweb [KNLZ07] and SENSEI [PBEV09]. The SENSEI project however does not provide support for dynamic changes in context as it relates to the discovery of new sensor sources, requiring the explicit modelling of the user-sensor relationships. However, the centralization of these approaches undermine their ability to scale well impacting on performance with respects to real time constraints. It further only considers proximity with respects to spatial measurements, excluding all other dimensions of context.

Schmidt, et. al. [SBC99] argued that context aware applications need to examine other dimension and expressions of context other than location when delivering context based experiences. With mobile applications and services benefiting from such an approach, we are need of new definitions and expressions of context proximity. Proximity that is derivable from all dimensions of context permitting applications less dependent on location to reason over more conceptual representations of distance; degrees of separation between entities as factor of their current context values.

This highlights the need for an approach permitting avenues for exploring the modelling of context information respecting all possible dimensions of context. Solutions indicating similarities among entities and supporting the deployment of services.
Consider the following scenario:

Tasha is a young woman living within a typical future urban environment where people are constantly on the move for business or pleasure. She carries a mobile device that continually seeks to define her current context and deliver services in response. Within such a future cityscape, she is exposed to numerous information points which maybe used to inform on her current context. A digital ecosystem capable of providing enough information in order to derive support for services wishing to effect changes or deliver experiences to a user based on her context. For her this includes audio devices, advertising video portals, internet connections or simply a range sensors reporting on temperature, humidity, lighting and capacity or even location, traffic and air quality.

Existing solutions such as [HMS+01], [BKr03] and [BHC06] could permit Tasha to discover such points. However, they would require the installation of additional devices. Firstly, devices attached to points around the cityscape or within buildings and secondly a device attached to each her. This solution has some advantages with respect to providing known anchor points, and partially negating the problems of GPS dead-spots. However, the user would be required to position himself within some spatial proximity of the device connected and perhaps motioning in order to initiate a proximity indicator to the supporting architecture. This would then create a connection between the user and some artefact; providing access to the information points available. With this approach, is required Tasha’s explicit intervention along with being dependent on her knowing the location of such points with which to synchronize. With a large population, this could become a nuisance with people queuing in order to be synced with the infrastructure, negating much of the progress made in realizing seamless ubiquitous computing. Any solution to deriving context within such a heterogeneous landscape must consider all indicators and be able to derive context information with minimal user interaction.

She could choose to remain dependent on solutions that derive proximity through the use of spatial locating techniques such as GPS, but would be subjected to their functional limits, such as being on a subway train. We concur with [SBG99] and [SDA99] and move away from the concept that physical location provides the overarching indicator of context and is required in order to enable useful context dependent services. Other work such as [PBEV09] would allow her to be connected to the cityscape but achieves this by using largely static determinations of what constitutes the information points connected to a presentity. This gets challenged however, if the city won the Olympic Bid, and an impromptu concert is being held in the city; the user would have no access to the ad hoc resources as [PBEV09] does not permit such additions, instead relying on a defined infrastructure; created, designed and instantiated by administrators. This further undermines the dynamic behaviour of the connections and people in the real world.

Tasha, however possesses context indicators hinting at his proximity with regards to other presentities within the digital ecosystem. She might have a GPS sensors, a physical proximity sensor or just a calendar event indicating his expected location.

Previous work concerning data mining within context centric architectures [KRZ+05] provided a means of deriving relationships between presentities and other entities or
identifying patterns in the information they contain. We however are not in search of data mining algorithms which require a collection of data capable of supporting inference. The dynamic nature of context information undermines this and results in solutions where data-mining is carried out over historical records on the presumption that it represents an indication of current context. In contrast, consider a user entering a city for the first time, his current context would display a change in patterns rendering all services based on his previous behaviour irrelevant. Such expectations of a context support system is not unusual in a world where people are becoming increasingly mobile and dynamic.

Schmohl and Baumgarten in [SB09] detailed an approach to deriving context proximity over multiple dimensions as a single value. Here, each dimension is mapped onto a map akin to a geographic map and the resulting coordinates used to calculate the Euclidean distances between any two points. They employed the use of dynamic distance limiters reflecting the preferences of the users as a means of detecting whether a threshold has been met or exceeded. Schmohl and Baumgarten however assumes all dimensions are in common on all presentities therefore providing an accurate derived value. The calculations in [SB09] also ignored the scales of each dimensions as is done with taking the simple Euclidean Distance. This produced distances that are not directly comparable outside of presentities having the same attributes on the same scales. The need therefore exists for methodologies capable of evolving and establishing localized sets of information points capable of answering a query concerning a user in real-time. Mandated are solutions that, in real-time, identifies and collates information sources considered to be within close proximity [HL01] to a user’s context and therefore are able to provide context information supporting applications and services.
3.4 Estimating Sensor Ranking

The concept of ranking documents or entities in order to locate important and relevant information, has long been central to the Internet. As we converge into an Internet of Things, the need arises again, to explore avenues for ranking and finding relevant sources of information. Such information, however, is no longer written in static documents or embedded in hyper-links. This new frontier of information is modified no sooner than it is created; remaining both fluid and relevant.

Within a future urban setting, there will exist a digital ecosystem capable of providing enough information in order to derive support for services wishing to affect changes or deliver experiences to a user based on some context. This includes audio devices, internet connections, video devices; a range of sensors including temperature, humidity, lighting and capacity sensors or even location, traffic and air quality sensors.

Consider the scenario of Omari, a 10 year old child with a smart device able to connect to, and derive representations of, context from these points in order to support some monitoring application or services. His mother has a child monitoring station at home which reports on Omari’s current context situation; i.e. is it too warm, too cold, raining, etc. He should therefore be able to connect to the most accurate and reliable sensor reporting a current outdoor temperature in order for his Mum to decide if she needs to request that he returns home because the temperature is too low.

Current solutions to exposing relevant and available sensors rely on the availability of fixed information points from which to derive indications of context. Omari could be connected to the SENSEI architecture [PBEV09]. He would, however, be provided with physically closest sensor, or the sensor connected to his infrastructure in order to derive a temperature value. Approaches such as [FPN+05] or [KNLZ07] could also provide Omari with sets of sensors based on his context, however their reliance on centralization would undermine their ability to do so within real-time constraints. While the addition of new sensors would be made available to Omari, they would not be recommended based on any metric outside of being a part of his domain or infrastructure. If he is in a room, he would like to be assigned the temperature sensor attached to the room.

In such a digital ecosystem, there will exist multiple sensors in the same locality being offered to users as potential sources of context information. A user, such as Omari, must be provisioned with sufficient information in order to, whether manually or autonomously, select the most suitable sensor for reporting his temperature.

One such element of information is the current ranking, the implied reputation of the sensors available. An indication of reputation could be derived from the behaviour of the users; continually connecting to and using the sensor as a source of context information. An accurate and reputable sensor, by any measure, would more likely be chosen and used over a sensor that is considered to be inaccurate and usually unavailable. For Omari, the temperature sensor attached to the building might not be the best sensor, as it might situated close to the heating radiator and reporting a reading that is several degrees higher than the actual temperature in the room. However a sensor temporarily placed in the room would not be made available by
which would first require it to be added to profiles and made available, which is not be feasible in dynamic environments. Therefore, there exists, a need to be able to identify changing patterns in user behaviour, such as most users not connecting to the existing sensor but to the alternative, i.e. a sensor ranking approach complementing the sensor proximity approaches mandated by [HL01].

Approaches to sensor ranking, such as [PBEY09], attempt to find sensors capable of answering a query at a given point in time. Such an approach is focused on the values from the sensor and not the sensor itself, the resulting ranking is only relevant to the query and disregards the remaining users within a system. In contrast to previous work, current ranking approaches such as Internet search engines consider the theory of connected things, however within the constraints of a relatively Internet. A document’s connectivity determines its relevance. This concept of ranking has been explored and used both in a centralized solutions [Goo10] as well as distributed solutions [ZYL05].

Centralized solutions such as Google index only a tiny portion, less than 10 billion of the estimated 550 billion pages, on the relatively static Internet [LLH+03] [ZYL05]. Any attempt to apply such a centralized solution to the ranking of sensors in an Internet of Things would be undermined by their ability to scale well. While improvements could be realized by increasing resources or by caching data, such an increase in resources requires considerable investment as network usage grows. In a future connected things society of over 50 billion devices, this would outstrip the current internet and its demands for computation. Considering these issues, [HO99] suggests that areas such as personalized PageRank calculation could only be realistically achieved using distributed approaches. The caching of data is also an approach to reducing computational overheads, however it puts a limit on the freshness of data and the determination of proximity values that cannot be guaranteed for accuracy or relevance.
3.5 Summary

This chapter summarized some of the theory and related work in the areas covered by this thesis. By means of exploring alternative and existing approaches, it provided further motivation for the need to explore solutions to creating context-centric architectures capable of provisioning, modelling and relating sensor and context information in a scalable and responsive manner. The need exists to enable distributed collections of context objects capable of being arranged into simple relevant subsets and be presented to applications and services in supporting of the realization context-based user experiences. Achieving this in a highly dynamic environment requires algorithms for comparing objects and providing metrics through which to achieve this organization. Additionally, this must be achieved over a scalable distributed infrastructure capable of accomplishing this with minimal overhead costs with regards to computing operations.
Chapter 4

Approach

The theory and related work detailed in Chapter 3 provided an understanding of the directions being explored towards realizing an Internet of Things. It mandated the need to enable distributed collections of context objects capable of being organized into simple relevant subsets and be presented to applications and services in supporting of the realization context based user experiences. An approach towards achieving this in a highly dynamic environment would require algorithms capable of comparing objects and providing metrics through which to achieve this organization. Additionally, this must be achieved over a scalable distributed infrastructure capable of accomplishing this with minimal overhead costs with regards to computing operations.

This chapter provides an introduction and overview of my contributions detailed between Chapters 5 and 8; comprising of a collection of objects organized into models and distributed over supporting P2P overlay. The overlay provides for persistence and messaging as well as a means of provision the supporting context information. It further details the application of context metrics towards finding and organizing related subsets of context information in end-points.

The architecture detailed in Chapter 5 was chosen as the enabling distributed architecture. This contrasts with existing solutions such as IMS which are centralized solutions dependent on 3G Internet and introduce reliability issues coupled with their general inability to scale well. The architecture detailed in Chapter 5 bears advantages over Cloud solutions which are reliant on DNS in order to reach web service portals, users and applications. Issues with DNS availability due to DoS attacks and configuration errors raises questions about its continued suitability and prompting research into Distributed Hash Table (DHT) based overlays as possible replacements [PMT06]. The solution detailed in Chapter 5 also bears advantages over brokering systems which often employ presence models that are unable to scale well.

Citing the advantages raised in Section 3.2 a distributed object oriented model as detailed in Section 6 is realized. This permits the organization of context information relevant to an entity or thing and be made available to applications and services
residing in end points. Such a model, however cannot be completely designed or constructed with \textit{a priori} knowledge while simultaneously reflecting the ad hoc nature of the interactions it seeks to reflect and support.

In approaching the problem of model construction, a \textit{schema entity} is introduced along with algorithmic approaches for determining and organizing such schemata. Firstly, a distance function is defined for measuring context distance or proximity among entities. This creates clusters of entities that with some degree of similarity by exploring all available dimensions of context information available. Such clusters may be personal clusters centric to an entity, or general clustering in support of knowledge discovery. Secondly, an algorithm is detailed for ranking points of information in an attempt to identify important and useful sources of context information, providing entities with the information needed to evaluate the reliability of context information sources as well as the resulting relationships established over this information.

\textbf{Summary}

This chapter summarized the approach realizing a solution towards the problem raised in this thesis. While each element of the thesis addresses a problem in the complete picture, they are essentially solutions independent of each other and solve problems that exist outside of this instantiation. An element such as the ranking algorithm could be applied to organizing context information sources on a traditional search engine, however when combined with the overlay described presents a more complete solution. These are detailed further in the successive chapters noting the contribution of each in realizing the overall aims and objectives of this thesis.
Chapter 5

An Architecture for the Provisioning of Context Information

Section 3.1 motivated, by means of current and existing work, the need for architectures capable of provisioning sensor information within real-time constraints. In response to this and the other requirements discussed, such a solution requires a combination of a scalable responsive overlay, an interoperable exchange protocol and a flexible and open naming scheme. This chapter examines my contribution with respects to enabling such an architectural solution, existing as a component of the general MediaSense framework (See Papers I and II). Here I investigated alternative P2P approaches for underpinning the overlay and selected P-Grid as the approach. I introduced the use of object-oriented databases for local persistence in an effort to improve performance and natively persisting context objects. I also introduced the TRANSFER and SET primitives as well as a distributed Context Storage on the overlay (See Paper VIII). I also introduced the approach to naming objects outside of raw sensor values and persisting these alongside sensor information for fast look-ups and name resolution.

5.1 The Overlay

The provisioning of context information must be supported by a robust and scalable overlay. Existing approaches to creating Internet scale overlays include projects such as Pastry [RDD01] and Chord [SMK+01], distributed lookup tables with varying
levels or performance and responsiveness. However, for this implementation P-Grid [ACMD03] was chosen as the underpinning overlay.

P-Grid was selected as the overlay for several advantages it offers through its implementation. The first advantage being that while it shares a common behaviour to DHT based implementation with respect to being able to index and locate information, it however realizes a distributed binary tree as illustrated in Figure 5.2. In contrast to DHTs, the key space is partitioned among all the nodes and organized into a binary tree structure with each node’s location determined by the binary bit string representing the set of values for which the node is responsible. With this, it preserves the ordering on data and natively enables the resolution of both specific key, substring and range queries without any pre-existing knowledge. This is achievable with at most the same message complexity of most DHTs and has proven performance of $0.5\log N$ versus $\log N$ for a Chord based implementation [ACMD03].
The non-deterministic distribution of keys coupled with self-organization, offers improved resilience in the dynamic environments which are expected to exist in a future Internet of Things as it permits a more flexible self-organization in response to a very dynamic set of information. This is complemented with redundancy for fault tolerance; multiple nodes are assigned to the same key partition and nodes hold references to multiple partition holders. As a relatively future proof overlay, it readily permits future extensions and modifications. The innate functionalities of P-Grid are exploited towards realizing two key properties of the resulting architecture.

Firstly, a P2P overlay is obtained on which to implement the DCXP protocol; a key requirement for the architecture. The overlay is responsible for the registration of context identifiers, providing a means of resolving the location/identifier problem. A UCI, detailed Section 5.4 is registered as an identifier for a context resource located on the overlay. The overlay is further exploited for routing the DCXP messaging primitives around the overlay, sending messages between nodes, resolving nodes responsible for a UCI or a range of UCIs as well as building and maintaining the overlay. The DCXP layer implements the DCXP primitives described in Section 5.2.

Secondly, benefits are gained from an indexing layer native to P-Grid, which provides for a distributed context storage implemented on top of the overlay. This is further described in Section 5.5.

A secondary result of using P-Grid is that such an implementation would be more suitably adapted for highly dynamic and flexible environments such as mobile networks which are underpinned by issues relating to heterogeneity and reliability. However, while the overlay permits users to make distributed sensor information available, this is not sufficient to support the interconnected and evolving Internet of Things. It is instead necessary to exploit the overlay as a building block and to rely on a protocol that focuses on data dissemination across interested nodes.

**5.2 The Distributed Context eXchange Protocol**

The Distributed Context eXchange Protocol (DCXP) is positioned within the MediaSense project (Paper II) as the protocol for enabling reliable application level communication amongst nodes within a P2P overlay network. DCXP provides for a means of exchanging information required to provision context information in a distributed scenario. As an interoperable protocol, aligned with the open implementation underpinning common Internet protocols; any device connected the Internet is capable of registering with the overlay and engage in the provisioning or utilization of sensor information.

The DCXP protocol is a SIMPLE [BLJ05] inspired protocol. The protocol contains fifteen primitives for sharing sensor information and interacting with actuators connected to and participating in the overlay. Table 5.3 provides an overview of each implemented or proposed messaging primitive within the protocol.
5.3 Context User Agent - CUA

A computer wishing to participate within the context provisioning overlay is only required to implement an instance of the CUA. The CUA permits the seamless exchange of context information among sources and sinks, as well as the interaction with actuators. Each CUA corresponds with a node on the virtual distributed tree described in Section 5.1. Each CUA further contains some persistence in the form of object oriented databases (OODB) along with an API for creating applications and services consuming and responding to sensor information. It further provides the entry point for registering and resolving a UCI or a query across the CS.

5.4 Universal Context Identifier - UCI

The exchange of sensor information requires naming schemes that enable us easily identify resources. In response to this, the UCI naming schema provides a URI [BLFM98] inspired naming schema with the following syntax:

\[ dcxp://user[:password]@domain[/path[?options]] \]

where \textit{dcxp} is the new URI scheme name and \textit{domain} is a Fully Qualified Domain Name (FQDN) relating to where the CI is located. The \textit{user} and \textit{password} arguments are optionally used as a means of authorization. The \textit{path} adheres to the context information name space hierarchy, permitting the organization and sorting of the items in a logical sense while \textit{options} facilitates further modifiers in the form of \textit{parameter=value} pairs.

An example of a fully qualified UCI adhering to this would be:

\[ dcxp://andeen.mccarthy@miun.se/weather/temp?unit=celsius \]

UCIs are registered with the CS and can be updated by the responsible CUA to reflect changes to locations, options and values. They are further indexed by the RI, permitting searching and range queries by applications and services.

5.5 Context Storage - CS

The architecture is supported by an underpinning Context Storage. In order to exploit the native characteristics of the overlay, the CS is integrated into the overlay. The key role of the CS remains that of resolving UCIs to physical end point addresses of the responsible nodes, behaving similarly to a dynamic DNS service. Here the substring and range queries enable the locating of resources without needing to know the fully qualified UCI.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGISTER_UCI</td>
<td>A CUA uses REGISTER to register the UCI of a CI with the CS.</td>
</tr>
<tr>
<td>RESOLVE_UCI</td>
<td>In order to find where a CI is located, a CUA must send a RESOLVE to the CS.</td>
</tr>
<tr>
<td>GET</td>
<td>Once the CUA receives the resolved location from the CS, it GETs the CI from the resolved location.</td>
</tr>
<tr>
<td>SUBSCRIBE</td>
<td>SUBSCRIBE enables the CUA to start a subscription to a specified CI, only receiving new information when the CI is updated.</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>The source CUA provides notification about the latest information to subscribing CUAs every time an update occurs or if asked for an immediate update with GET.</td>
</tr>
<tr>
<td>SET</td>
<td>Used to manipulate the behaviour of actuators.</td>
</tr>
<tr>
<td>TRANSFER</td>
<td>Used to relocate the responsibility of an information source to another node.</td>
</tr>
<tr>
<td>ANON_REGISTER_UCI</td>
<td>Used by a CUA in order to anonymously register the UCI of an information source with the CS layer.</td>
</tr>
<tr>
<td>ANON_RESOLVE_UCI</td>
<td>This is used in order to anonymously resolve the location of a UCI within the overlay.</td>
</tr>
<tr>
<td>ANON_GET</td>
<td>This is used to retrieve a value from a UCI where the requesting node wishes to remain anonymous.</td>
</tr>
<tr>
<td>ANON_SUBSCRIBE</td>
<td>This permits a CUA to anonymously subscribe to an information source, receiving new information when it is available.</td>
</tr>
<tr>
<td>ANON_NOTIFY</td>
<td>The source CUA provides notification about the latest information to subscribing CUAs every time an update occurs or if asked for an immediate update with GET.</td>
</tr>
<tr>
<td>ANON_DIALOG_RESOLVE_UCI</td>
<td>Similar to an ANON_RESOLVE_UCI, except that both the requester and responder are anonymous.</td>
</tr>
<tr>
<td>ANON_DIALOG_GET</td>
<td>Similar to an ANON_GET, except that both the requester and responder are anonymous.</td>
</tr>
<tr>
<td>ANON_DIALOG_NOTIFY</td>
<td>Similar to an ANON_NOTIFY, except that both the requester and responder are anonymous.</td>
</tr>
</tbody>
</table>
Additionally, the CS stores the current value associated with a resource, providing for a simple and quick way to retrieve the latest sensor information without needing to contact the responsible node which maybe offline. For a resource consisting of multiple dimensions such as GPS coordinates, each dimension is stored separately. We separate dimensions to further enable independent queries over any value constituting a context information source. An end point is then free to reconstitute and compare the entire n-dimensional value or any valid subset.

We further benefit from the overlay’s order preservation and range query properties, permitting the acquisition of useful data in support of context metrics or similarity functions. Thus providing more a robust solution in a heterogeneous network environment.

5.6 Persistence

The CS along with the DCXP protocol enables distributed access to only current context information. Any new values reported in connection to a UCI or context dimension supersedes the current value. This prevents applications from using expired or stale data. Historical information was previously persisted to a centralized relational database by the nodes. The dynamic nature of context information mandates the storage of historical information, both on the values and the actual context objects. However storing versions of information as different entries on the overlay, would undermine its performance as a direct result of the vast number of attributes persisted each node; this would be increased by a factor equal to the number of versions.

We therefore solve this problem by using an additional layer; the persistence layer. This consists of a collection of localized Object Oriented Databases residing at each CUA. Each node stores the set of objects that are created locally by the application and services co-located with the CUA. They further store a collection of objects that are being used by these applications and services but originate at a remote node within the overlay. We also persist all observed values for each context dimension permitting temporal views of data evolution, trend and pattern discovery.

Before an object is persisted locally, an attempt is made to persist it on the overlay. If this is successful, then the object being stored is guaranteed to be globally unique and is then stored locally with its UCI on the overlay. If this is not successful, the UCI is being stored already exists and the persistence operation fails. This ensures that locally persisted objects are always globally unique. All objects that are being used by applications and services local to the CUA are stored locally, these contribute to some local schema (Section 6.3) being used at the node.
5.7 Resource-Constraint Devices

The provisioning of sensor information in a pervasive environment involves interaction with nodes on resource constraint devices. Such resources include processing, battery and unpredictable network performance amplified by devices provisioned with 3G Internet access by means of mobile radio networks.

In response to this, a Mobile DCXP Proxy (MDP) is implemented as a means of mediating between the more stable P2P backbone, shielding it from the more unreliable mobile nodes susceptible to packet losses and unavailability inherent in radio networks. The MDP is realized using two components: a remote CUA (rCUA), residing on a more stable node and fulfilling the requirements of a CUA handling all overlay functionality on behalf of the mobile device. On the mobile device, a Limited CUA (lCUA) is implemented providing CUA like service to the mobile device without fully participating in the overlay. Its requests for sensor information are forwarded to the rCUA which returns results. The rCUA further handles all requests for information from sensors co-located with the lCUA forwarding this information to the lCUA when it is available. This presents a seamless service to both the overlay and the resource constraint device.

5.8 Evaluation

In this chapter I presented a response to the problem of being able to provision sensor information and support context information models in a scalable manner. The DCXP centred architecture liberates this provisioning from information portals dependent on DNS and introduces a DHT-like approach capable of self-organization in response to the data being persisted. The DCXP architecture achieved early results (Paper[1]) which on par with UDP for the provisioning of context information by means of a peer-to-peer overlay. Here, this overlay provides us with the ability to create models from objects that are persisted locally but made available through a global addressing and look up scheme, a UCI. Object/location pairs are resolved without a dependency on a centralized server.

One issue that arises with my approach is that we are reliant on resource sharing which might be limited in the devices which constitute the majority on an Internet of Things. Currently, this is not a real issue but it could arise as users become increasingly mobile and resulting in bottle-necks where rCUAs are undertaking resource intensive operations offloaded by the lCUA. However as resources increase in mobile devices to the extent multi-core processing and storage capacities rivalling laptop computers, the lCUA/rCUA approach could be made redundant with full CUA nodes being executable on mobile devices.
5.9 Summary

In response to this the requirements raised in Section 5.11 with regards to a solution comprising of a scalable responsive overlay, an interoperable exchange protocol and a flexible and open naming scheme, this chapter detailed my contributions with respects to enabling such an architectural solution. I introduced P-Grid as the overlay of choice along with the use of object-oriented databases for local persistence in an effort to improve performance and natively persisting context objects. I also added the TRANSFER and SET primitives as well as a distributed Context Storage on the overlay and using UCI’s for naming objects outside of raw sensor values. These objects I also persists alongside sensor information for fast look-ups and name resolution.
Chapter 6

Distributed Models Supporting Context Services

Section 3.2 discussed approaches to presenting and modelling context information. In contrast to the simple provisioning of sensor information discussed in Section 3.1, it motivated, by means of current and existing work, the need for a new approach to modelling context information. It further explored how such an approach must be scalable over the distributed solutions motivated by Section 3.1.

This chapter examines my contribution with respects to enabling such a solution, and its ability to be incorporated into the wider Internet of Things. I introduce a method for distributing an object-oriented model for context awareness over the architecture detailed in Chapter 5 (See Papers V and VIII). For this I utilize the overlay for locating and retrieving context objects identified by a UCI described in Section 5.4 and persist these objects by utilizing the Context Storage described in Section 5.5 as well as the persistence mechanism detailed in Section 5.6. Additionally I introduce the concept an an object schema as a means of organizing and maintaining collections of related objects in end points over the distributed architecture.

6.1 The Context Information Integration Model

The modelling of context information within the MediaSense framework is realized through the Context Information Integration Model (CII) as detailed in [BLKW10]. This model describes an entity-predicate-entity triple implemented in an object-oriented framework. Such a model, as illustrated in Figure 6.1, is similar in concepts to the semantic web approaches, however it remains advantageous with regards to the time taken to traverse and reason over an object-based model. The main concepts of the CII model are summarized in Table 6.1. Such a model provides a way to represent the relationships and interactions among the connected things within an Internet of Things, where things can range from sensors and actuators to virtual information.
sources such as social networks, media, people and infrastructure.

The CII model is a collection of objects with defined relationships. As an object-oriented approach, it inherently gains the properties of class inheritance and as such can be extended with new sub-concepts of Entity and Information-Source. This integration would be made possible by adaptive software techniques such as a combination of computational reflection, automated compilation and dynamic class loading. Agents, applications and services reside above and use the meta-model as a source of data and deriving context information.

The model presented in this thesis, deviates from the originally model conceived and implemented in [DLKW10]. The model, in order to support a more applicable Wireless Sensor Actuator Network (WSAN) scenario, has the concept of Information Sources superseded by Information Points. Actuators now become Information Sinks with the following reverse properties of Information Sources:

- Comparing input values: a Fahrenheit value could be passed into the end point and compared with the threshold value of the actuator
- Representations and translations: exposing multiple representations for accepting and translating input, an actuator implemented in Centigrade could accept temperature settings in Fahrenheit
6.2 The Distributed Context Information Integration Model

Distributing this model, while achieving and sustaining real-time targets, mandates a need to address three main problems. Firstly, the need for a scalable distributed
architecture for routing context information among nodes in real-time; secondly real-time querying and indexing infrastructure for locating entities and information across a distributed architecture. Thirdly, a means of constructing and manipulating complex object-oriented context models representing the highly dynamic real world interaction of presentsities. A support for this is illustrated in Figure 6.2. The first two are resolved by the DCXP and Resource Index layers respectively.

Implementing the CII elevates the CUA to an intelligent context node with the ability to initialize, store and manipulate context models required to support applications and services being executed at an endpoint. Context models are persisted and built in object-oriented databases co-located with the CUA at each node.

Within a DCII, an object representation of a sensor or actuator is defined. The object is stored locally in an OODB and also indexed in CS. The same happens for all entities including presentsities. The application or service responsible for the creating and maintaining the object further adds relationships to its schema, define translation rules and access permissions.

When a sensor or actuator is added to the DCXP layer it is given a UCI of the format:

\[
\text{dcxp://user[:password]@domain[/path[?options]]}
\]

The original DCXP interactions remain. The local CUA implements an object representation of the sensor as described by the CII model and stores it in the local OODB. Each object is assigned a UCI of the format:

\[
\text{dcii://user[:password]@domain[/path[?options]]}
\]

The object UCI is persisted and index in order to facilitate real-time resolution of UCI to object queries. The object is then serialized and indexed by the RI layer for supporting searching and browsing.

An application requiring use of a sensor implements a reference to a sensor object. The local object layer is then responsible for resolving the UCI, fetching the object description, initializing the object and translating this into the \textit{GET} or \textit{SUBSCRIBE} primitives of the underlying sensor. It further maintains this relationship until the sensor is no longer required. This interaction is local to the application and its CUA. The CUA responsible for the sensor object is not required to participate in this relationship.

Such object-application relations are straightforward, however when an application requires the context-model for a presentsity in order to deliver some service or user experience, the model then introduces more complexity. Here, we utilize the context schema described in Section 6.3.
6.3 Context Schemata

A Presentity as regarded by [Chr02] is: An entity that possess and is defined by its presence information. They are unique in behaviour from other entities such as sensors or actuators. In response to this, I introduced the concept of a Schema Entity in Paper [VIII]. This is attached to a presentity and describes the current model of sensors and actuators that provide context information supporting the presentity. In this way the watchers, regarded by [Chr02] as the entities interested in a presentity's presence, may have access to a defined real-time picture of all the information points related to a presentity. It can then choose which sensors to use in order to deliver its services.

Schema entities expose a PUBLISH/SUBSCRIBE interface. This approach permits avoiding having to synchronize large datasets distributed around the architecture. Watchers can therefore subscribe to a schema and be notified as it changes. There is no need to issue queries to nodes or databases or for watchers to be concerned with checking for updated presence information.

The CUA local to a presentity maintains its schema, adding or removing sensors or other information sources that contribute to its presence profile. Where an application requires the Information Points relating to a person, a subscription is made to the CUA local to the application. This liberates the CUA hosting the presentity from maintaining resources capable of supporting all the services connected to it. The schema is then resolved to a DCXP PUBLISH/SUBSCRIBE with the concrete sensor/actuator sources. The application’s local CUA is then responsible for maintaining and updating the local representation of the schema relative to the application’s requirements. With this loose coupling, applications can ignore schema changes that do no impact on their performance, e.g.: an application that requires location information may ignore schema changes that adds new temperature sensors but would update the context-model if a new location sensor is added to a presentity. Such an approach is beneficial in resource-constrained devices where only a subset of the schema may be implemented.

The schema entity detailed in Section 6.1 introduces support for the dynamic behaviour of context information. We need to maintain a model that continually describes the current situation of a presentity. This we regard as schema evolution: the continual adding or removing of information sources that are available to a presentity and subsequently its watchers. Model schema evolution will take place progressively, continually deriving its new state from application level interaction. Applications such as user agents can negotiate the addition of new sensors or actuators to a presentity’s schema.

The watchers (applications, services, etc.) can SUBSCRIBE to the schema object and receive notifications of any schema changes.

The evolution of the schema provides for personal preferences with regards to this problem. Schemata are seen as being infinitely composable and reusable such that a new schema may be constructed over existing schemata. Such an example might be the need to express the collaborative context of all the occupants in a room in order...
to derive an accurate context-model for the room itself. This however can be limited by the fact that the schemata are time constrained and encapsulate the composition of a subset sensors attached to a presentity $P$ over time $t$.

The process of evolving of a schema is triggered by the presentity itself establishing a recursive dependency where, a schema expresses a new context-model which is used by the applications. The applications through defined dynamic interactions allow for resource discovery which may in turn trigger the evolution of the schema. Applications dependent on a presentity’s context do not need to discover and negotiate with sensors directly. Information sources will be seamlessly added or removed modifying the schema being used.

### 6.4 Evaluation

The Distributed Context Information Integration Model (DCII) presented in this chapter, is an approach to enabling context models over a distributed architecture. This DCII model is created as an object-oriented ontology abstracting from the underlying sensor implementations while presenting an expressive collection of objects to applications and services. I offer an improvement over solutions such as IMS and Mobilife, which are dependent on web service portals on the Internet in order to deliver information models and schemata to applications and services. By realizing this model on a distributed overlay, this dependence is removed with a node simply needing to participate in an overlay in order to gain access to the information available.

By enabling the derivation of context information from heterogeneous sources over the Internet, I offer an improvement to other distributed object approaches such as The Hydrogen Project which is restricted to context information availably locally in the end devices. Other distributed models such as The Context Toolkit are inferior with regards to expressiveness as it stores models as attribute/value enabling little room for modelling the dynamic behaviours and interactions on the Internet of Things.
The introduction of object schemata provides a means of identifying and creating information subsets related to a user. By enabling continual evolution supported by a *publish/subscribe* interface, dependent applications and services are not required to make unnecessary calls to a server or endpoint incurring additional overheads as would be the case with a centralized or broker based architecture.

### 6.5 Summary

This chapter explored the main contribution with regards to the creation of models capable of representing the complex interactions of entities within a context-centric architecture. I detailed my contributions with respects to enabling such a solution, and its ability to be incorporated into the wider *Internet of Things*. I introduced a method for distributing a object-oriented model for context awareness over the architecture detailed in Chapter 5. For this I utilize the overlay for locating and retrieving context objects identified by a UCI described in Section 5.4 and persist these objects by utilizing the Context Storage described in Section 5.5 as well as the persistence mechanism detailed in Section 5.6. Additionally I introduce the concept an an object schema as a means of organizing and maintaining collections of related objects in end points over the distributed architecture. Further reading on this contribution can be found in papers: [I] [V] [VII] & [VIII]
Chapter 7

Estimating Context Proximity

The ability to derive models representing a presentity’s current context, underpins our ability to enable applications and services tailored on user experience. Section 3.3 discussed existing solutions enabling derivations of context proximity or solutions permitting the delivery on context aware services in response to some proximity factor. Such solutions, as discussed are often undermined by their dependencies on centralized architectures limited with respect to scalability.

In response to these shortfalls, this chapter details my contribution as an algorithm for estimating the context proximity among presentities (See Paper IX), enabling complete schemata of entities as one traverses the vast and dynamically connected things infrastructure. This addresses the problem of being able to group relevant subsets of context information in end points by permitting these end points to describe the subsets and discovering only information that satisfies their requirements. Additionally, the gossiping algorithm described as a part of my contributions in Paper VI is extended to be distributed on top of the overlay described in Section 5. This is in order to optimize the ability to navigate and create schemata of entities (See Paper VIII) within a distributed things infrastructure while addressing the problems with computational overheads.

7.1 Overview

Users or other presentities interacting within active context networks establish associations which are highly volatile and dynamic. However, while these associations exist, they provide us with implicit information on the state of a presentity and its relation to other presentities existing across the network. Consider Figure 7.1 with two presentities $P_1$ and $P_2$ as well as two sensors $S_1$ and $S_2$. While both presentities are connected to $S_1$, they derive an implicit connection to each other. If both presentities each has a single context value, $S_1$, then a suitable deduction would be that they are both experience the same definition of context.
Consider the alternate scenario:

- $P_1$ is connected to and derives a current context solely from $S_1$
- $P_2$ is connected to and derives a current context solely from $S_2$
- $S_2$ has a similarity of $d_1$ to $S_1$

The likelihood that $P_1$ and $P_2$ are sharing a similar context is directly related to $d_1$; the proximity or similarity between each context value. To determine this, one could simply find the difference between both values as an indicator of the similarity between both presentities. This however raises two issues: to what degree does $P_1$ or $P_2$ consider $S_1$ or $S_2$ respectively as an indicator of its current context, and does $P_1$ or $P_2$ have a preference with regards to what it considers to be similar. In addressing the problem of determining context proximity, I considered these issues and introduce an approach that reflects this.

I explore these underlying connections among presentities by constructing localized weighted graphs reflecting the degree of connection between $P_1$ and all other entities within a given range. The degree of similarity between each presentity being the weights on the edges of the graph. I further assume each node in the graph to be in a Markovian state such that in randomly walking across the graph, the probability of transiting from $P_1$ to $P_2$ is entirely dependent on the state of $P_1$ and the weighting on its edge $d_1$ as the amount of effort required to make the transition. This assumption permits me to derive the degree of similarity between $P_1$ and $P_2$ by disregarding all other nodes’ regards for $P_1$. Additionally, by considering each presentity’s context to be in a Markovian state, I am can make guarantees on the subsequent states and connections in that state, independent of the previous context state. This I intend to address in future work.
In order to build a relevant subset of presentities relative to $P_1$, I first permit $P_1$ to determine what is to be considered relevant and similar by explicitly stating the size of the range, $R_1$. The multi-layered architecture detailed in Chapter 4 permits the definition of $R$ with such clarity as: \textit{people within 3 km with a temperature less than 5°C difference}.

The assumption is made that each information point is capable of providing context information in some measurable form, either discrete or continuous. This however, acknowledges the existence of information points where such measurable representations of context might not be available or are more difficult to obtain. Such a situation could occur when comparing appointments on calender, where a simple \textit{true/false} would suffice in comparing current meeting availability or if the light in an apartment is bright enough for a person to read. Previous work in [DLKW10] concerning the creation of object models for context representation, addressed this by permitting information points to implement comparator interfaces. This would allow us to compare two values; obtaining either a boolean comparison or some other measurable value, as dictated by the implementation of an information point, representing the distance between two values.

For a dimension $S_1$ attached to $P_1$, with a value of $V_1$, a query can be made in order to obtain information points bearing context values to some degree but within the limits of $R_1$, the maximum range of $S_1$.

This permits the creation of a cluster of information points, each with a context value $V_i$ such that:

\[(|V_1 - V_i|) \leq R_1 \tag{7.1}\]

Here, $V_1$ is the current value of $S_1$ and $V_i$ is the current value of $S_i$. Therefore within a domain $D$, $S_1$ obtains a set of related sensors $S$ at time $t$ such that:

\[G_t = \{S : S \in D_t : (|V_1 - V_i| \leq R_t)\} \tag{7.2}\]

This is a dynamic set of information points with respect to $P_1$ and its context dimension $S_1$, that continually evolves to reflect the addition or removal of sensors with respect to their current values. When new information points obtain values within $R_t$, they are added to $G$, while they are removed when their values exceed $R_t$.

The algorithm further considers that not all instances of $S_i$ lie within the same proximity to $S_1$. This implies that $S_1$ shares a closer context with some members of $G$ and subsequently those members must be given a higher preference with regards to any context dependent application or services wishing to find context information points in support of delivering some optimal user experience. Using the distance $|V_1 - V_i|$ would not be a reliable indicator of such relevance, since the scales could be
different for each sensor $S_i$ attached to $P_1$. I normalize these values as a function of
the value with respect to the scale and the distance from $S_1$ as follows:

$$CS_{S_i} = f(S_i) = (1 - |V_{S_i} - V_{S_1}| \cdot R_{S_1}^{-1})$$ (7.3)

where:

$$0 \leq CS_{S_i} \leq 1$$ (7.4)

A value of 0 being at the edge and not very similar to $S_1$ while 1 being identical to $R_{S_1}$. This is a continuous range of values with $CS_{S_i}$ being a value between 0 and 1.

This value is useful for us for two reasons, firstly it can be used to apply a weighting to the edges connecting $S_1$ to $S_i$ and subsequently the edges connecting to $P_1$. Secondly, it can be used to calculate the average similarity $GCS$ of the group $G_{S_1}$. This is expressed as follows:

$$GCS_{S_1} = \sum_{i=0}^{n} (1 - |V_{S_i} - V_{O}| \cdot R_{S_1}^{-1})$$ (7.5)

where:

$$0 \leq GCS_{S_1} \leq 1$$ (7.6)

A value closer to 0 being not well connected and 1 being connected to a set of
sensors with similar context values. This represents an indicator of the connectivity of $S_1$ with respect to its current context value, and an indicator of the probability that there exists good connections to presentities sharing a similar context to $P_1$. Assuming $P_1$ treats each sensor equally, an application interested in enabling some service based on $P_1$’s context can begin by exploring the groups with the highest ranking attached to $P_1$. Those groups would more likely contain links to entities with a context similar to $P_1$. Figure 7.3 illustrates such a final cluster of information points around a single dimension.

### 7.3 Multi-Dimension Proximity

Section 7.2 details an approach to deriving context proximities in a single dimension, further creating clusters of information points within a target range. In reality however, such a relationship would more likely be represented over multiple dimensions of context as illustrated in Figure 7.3. Here it is derived that the two presentities posses some degree of context similarity owing to the underlying sensors supporting their context being within some proximity.

In deriving a value representative of this overall proximity, I extend the simple Euclidean Distance. Calculating the simple Euclidean distance measures the distance
between two points over their set of attributes and is widely used in distance functions for clustering and data mining. However the Euclidean distance does not consider the scales on which the distance is measured, which raises a problem when attempting to compare the distance among presentities. In such scenarios 5°C does not convey the same meaning as 5km. Additionally it does not consider the intent of the user where for a user in search of a hospital 5km could be very close, but for a user searching for a pub 5km could be very far away. I therefore introduce an approach that considers both of these. Firstly, the user decides the range, \( R \), as shown in Section 7.2. I address this by normalizing the Euclidean distance between both presentities over the maximum distance allowed for the current user considering all its values for \( R \). Additionally I permit the specification of weighting, \( W \), for each dimension permitting an application or service to adjust this as required, reflecting the importance of each dimension as a definition of context relative to another presentities or towards the fulfilment of the services it requires. An application interested in finding a person for conversation might be more interested in their online status than their geographic distance. We begin by calculating Presentity Similarity (PS) as:

\[
PS_A = 1 - \left[ \left( \sum_{i=0}^{n} W_i (S_1 - S_i)^2 \right) \times \left( \sum_{i=0}^{n} W_i (R_i)^2 \right)^{-1} \right] \quad (7.7)
\]

The derivation of \( PS \) as show in Equation 7.7 assumes that all dimensions or \( S_i \) are common to all presentities and discards all dimensions that are not common. Simply using a null value for a missing dimension does not suffice, since this assumes the value is known to be zero, this is in fact much different from the actual scenario which would be that the value is unknown. In addressing this problem, I determine the set similarity between \( P_1 \) and \( P_i \). For this I take the Jaccard Similarity [DKCM10] between each set, their context schemata as discussed in Section 6.3.

Consider that \( P_1 \) is connected to four sensors, each with a cluster of sensors within its given proximity, such that the set of sensor clusters of \( P_1 \) would be:
$P_1 = \{G_w, G_x, G_y, G_z\}$

$P_2$ is connected to three sensors, each with a cluster of sensors within its given proximity, such that the set of sensor clusters of $P_2$ would be:

$P_2 = \{G_w, G_x, G_y, G_m\}$

Based on this, the similarity between the schemata of $P_1$ and $P_2$ is calculated as:

$$PS_B(P_1, P_2) = \frac{|P_1 \cap P_2|}{|P_1 \cup P_2|}$$

$$PS_B = \frac{|G_w, G_x, G_y|}{|G_w, G_x, G_y, G_m, G_z|} \quad (7.8)$$

This permits the comparison of values that cannot easily be measured such as favourite colour, mood, etc. In these expressions of context, it is not possible to perform discrete distance measurements or limit proximity based on this, however one can provide mechanisms for grouping together similar values which might equate to a user saying: *I like red, but pink, and purple are also acceptable alternatives.* While it is non-trivial to calculate a measurable distance between this set of values, treating it as a set of information points supporting a presentity permits a comparison to another set. If a presentity was comprised entirely of such values, one could still derive a measurement of distance based on the grouping of these values and finding the degree of similarity between presentities. Here an application could define which dimensions of context must be taken into consideration when calculating similarity,
such that for an application only interested in distance and temperature only, the corresponding equation would be:

\[
PS_B(P_1, P_2) = \frac{|G_{w}, G_x|}{|G_{w}, G_x|}
\]  

(7.9)

or for distance, temperature and humidity, where \( P_2 \) had only two dimensions:

\[
PS_B(P_1, P_2) = \frac{|G_{w}, G_x|}{|G_{w}, G_x, G_y|}
\]  

(7.10)

The PS is adjusted to reflect the Jaccard Similarity between both schemata in order to reflect a more accurate representation of PS and to account for the missing dimensions between \( P_1 \) and \( P_i \). The final value for PS is determined as:

\[
PS = \left( \frac{|P_1 \cap P_i|}{|P_1 \cup P_i|} \right) \times \left[ 1 - \left( \left( \sqrt{\sum_{i \in P} W_i (S_i - \bar{S_i})^2} \right)^{-1} \right) \right]
\]  

(7.11)

and the presentity distance \( PD \), to be:

\[
PD = 1 - PS
\]  

(7.12)
Algorithm 1 Finding and Ranking Relevant Information Points

loop
{at the local node}

for all information points attached to $S_1$ do
  determine maximum proximity value
  issue a range query for information points within proximity
end for

for all results, $S_i$ received do
  calculate proximity between $S_1$ and $S_i$
  if proximity $\leq$ max then
    add information point to list
    calculate the ranking $r_{S_i}$
    attach $S_i$ to $S_1$ with a degree of $r_{S_i}$
    calculate and update group ranking
  end if
end for

{at the remote nodes}

for all range queries received do
  if there are local information points matching query then
    return information points to originator, $S_1$
    create a standing query, notifying the originator of this
  else if there are known peers with points matching this query then
    forward the query to each peer
  else
    ignore query
  end if
end for
end loop

7.4 Realization in Distributed Architecture

The algorithms detailed in Sections 7.2 and 7.3 would gain the most suitable performance when implemented across the architecture detailed in Chapter 5. Within such an architecture, an application may not necessarily reside locally to the nodes it is trying to support nor any of the nodes on which a context neighbour resides. Here, an application may fulfil this by subscribing to the presentity’s schema and using this the means of deriving the context metrics required.

Our distributed solution, would rely on the DCXP architecture detailed in Chapter 5 for messaging support, and implement a distributed gossiping algorithm.

The distributed gossiping algorithm was detailed in publication VI as a lightweight peer-to-peer algorithm for organizing entities into small dynamic groups based on
some indicator value. The main aim, was to derive the ability to maintain groups that were centred around an entity according to the preferences of the entity expressed as a known measurable value. Such organization was unstructured and occurred as values changed with respect to changes in the entity itself or changes to the entity’s affinity to the value.

Within this architecture, I eliminate all dependencies on centralization by depending on the the resource index of P-Grid as a means of locating initial starting nodes. A presentity issues, on joining, a search for other presentities matching its proximity criteria, it retrieves an initialization list with which to start probing for entities within close proximity, recursively doing so to locate all known entities matching its criteria.

By this principle, an application wishing to find entities close to \( P_1 \) would accomplish this by first locating some initial nodes with a context proximity \( X_1 \) of \( S_1 \). Obtained from this query is a list of matching entities and a series of replicated running queries at each peer with the following constraints:

1. The peer is responsible for a sensor fitting the criteria of the search
2. The peer is responsible for a sensor \( S_i \) with a range such that the set of sensors fulfilling the query from \( S_1 \) would be a subset of a query from \( S_i \)

Each peer is then required to:

1. Forward the sensors matching the standing query to \( S_i \)
2. Forward the query from \( S_1 \) to any node it encounters that matches 1 & 2 above

This algorithm is only executed at a remote node \( S_i \) while \( S_1 \) maintains a relationship with \( S_i \). When this is not the case, the peer responsible for \( S_i \) cancels the standing query and no longer forwards it. With this approach we minimize the number of sensors being forwarded in response to an expired query. Further, since the indicators of context such as temperature or location are likely to change gradually, applications requiring a new set of nodes in response to changed context would benefit from \( S_i \) forwarding the query to current matching nodes. This, as the query would still likely be valid for a subset of these nodes or nodes that are within their groups. The algorithm is summarizes in Algorithm 1.

### 7.5 Verification

The algorithm detailed in Section 7.11 was tested to verify that it is capable of calculating a proximity value for a set of presentities. For this, a set of ten presentities were assigned four different context dimensions. The number of dimensions were identical for each presentity in order to obtained a common comparison for all presentities. For this test, one presentity was chosen as the home node \( P_1 \) and the context similarity and context distance were calculated for each instance of \( P_i \) with reference to the current context of \( P_1 \). The interest ranges, \( R_{S1} = 15 \), \( R_{S2} = 20 \), \( R_{S3} = 25 \),
$R_{S4} = 20$. These values were used to calculate both the Proximity Similarity $PS$ and the Proximity Distance $PR$. The results are shown in Table 7.1.

Table 7.1: Verifying Context Proximity - Results 1

<table>
<thead>
<tr>
<th>Presentity</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>PD</th>
<th>PDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>35</td>
<td>18</td>
<td>16</td>
<td>17</td>
<td>0.000</td>
<td>1.000</td>
</tr>
<tr>
<td>$P_2$</td>
<td>32</td>
<td>23</td>
<td>19</td>
<td>15</td>
<td>0.169</td>
<td>0.831</td>
</tr>
<tr>
<td>$P_3$</td>
<td>34</td>
<td>24</td>
<td>27</td>
<td>18</td>
<td>0.310</td>
<td>0.690</td>
</tr>
<tr>
<td>$P_4$</td>
<td>15</td>
<td>21</td>
<td>29</td>
<td>25</td>
<td>0.624</td>
<td>0.376</td>
</tr>
<tr>
<td>$P_5$</td>
<td>25</td>
<td>17</td>
<td>31</td>
<td>24</td>
<td>0.477</td>
<td>0.523</td>
</tr>
<tr>
<td>$P_6$</td>
<td>26</td>
<td>19</td>
<td>33</td>
<td>32</td>
<td>0.601</td>
<td>0.399</td>
</tr>
<tr>
<td>$P_7$</td>
<td>27</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>0.793</td>
<td>0.207</td>
</tr>
<tr>
<td>$P_8$</td>
<td>31</td>
<td>32</td>
<td>23</td>
<td>31</td>
<td>0.526</td>
<td>0.474</td>
</tr>
<tr>
<td>$P_9$</td>
<td>18</td>
<td>33</td>
<td>22</td>
<td>19</td>
<td>0.579</td>
<td>0.421</td>
</tr>
<tr>
<td>$P_{10}$</td>
<td>15</td>
<td>23</td>
<td>31</td>
<td>18</td>
<td>0.628</td>
<td>0.372</td>
</tr>
</tbody>
</table>

The results are indicative of the context similarity of each presentity with respect to the current context of $P_i$. As expected, the resulting values provide a distance and similarity measurement relative to the interest area of $P_i$ and value $R$ considered by $P_i$ to be the relevant scope required for realization of any applications or services.

Secondly, the algorithm was verified with with regards to its ability account for missing or incomplete dimensions of context, enabling it to achieve distance and similarity calculations in the absence of dimensions on either $P_i$ or any $P_j$. In such a scenario, the algorithm is expected to produce values for context similarity that are a function of the Jaccard Similarity of both sets. The value for presence similarity should never exceed the Jaccard Similarity. These results are shown in Table 7.2.

Table 7.2: Verifying Context Proximity - Results 2

<table>
<thead>
<tr>
<th>Presentity</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$PS_1$</th>
<th>$PS_2$</th>
<th>$PS_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>35</td>
<td>18</td>
<td>16</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$P_2$</td>
<td>32</td>
<td>23</td>
<td>19</td>
<td>15</td>
<td>0.169</td>
<td>0.831</td>
<td>0.383</td>
</tr>
<tr>
<td>$P_3$</td>
<td>34</td>
<td>24</td>
<td>27</td>
<td>18</td>
<td>0.310</td>
<td>0.690</td>
<td>0.483</td>
</tr>
<tr>
<td>$P_4$</td>
<td>15</td>
<td>21</td>
<td>29</td>
<td>25</td>
<td>0.624</td>
<td>0.376</td>
<td>0.096</td>
</tr>
<tr>
<td>$P_5$</td>
<td>25</td>
<td>17</td>
<td>31</td>
<td>24</td>
<td>0.477</td>
<td>0.523</td>
<td>0.299</td>
</tr>
<tr>
<td>$P_6$</td>
<td>26</td>
<td>19</td>
<td>33</td>
<td>32</td>
<td>0.601</td>
<td>0.399</td>
<td>0.319</td>
</tr>
<tr>
<td>$P_7$</td>
<td>27</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>0.793</td>
<td>0.207</td>
<td>0.124</td>
</tr>
<tr>
<td>$P_8$</td>
<td>31</td>
<td>32</td>
<td>23</td>
<td>31</td>
<td>0.526</td>
<td>0.474</td>
<td>0.209</td>
</tr>
<tr>
<td>$P_9$</td>
<td>18</td>
<td>33</td>
<td>22</td>
<td>19</td>
<td>0.579</td>
<td>0.421</td>
<td>0.047</td>
</tr>
<tr>
<td>$P_{10}$</td>
<td>15</td>
<td>23</td>
<td>31</td>
<td>18</td>
<td>0.628</td>
<td>0.372</td>
<td>0.088</td>
</tr>
</tbody>
</table>

Each calculation from $PS_1$ to $PS_4$ progressively reduces the number of values being compared from 4 to 2. The context context similarity was correspondingly reduced as was expected. The implication of this reduction being that the similarity is reduced.
to account for the dimensions of context that are unknown and as such provides an incomplete picture as to the similarity of the two presentities being considered. By simply stating which dimensions are to be considered when the calculation is being undertaken. The resulting action, being that the extra dimensions are not taken into consideration by the algorithm.

I performed a similar calculation but adjusted the weighting on the first dimension to test the ability of the algorithm to reflect the importance of a dimension in calculating context proximity. The results are shown in Table 7.3. The value for $PS_2$ is the new value while $PS_1$ is the original value with no weighting added. This verified that I am able to adjust the weightings to reflect the requirements of users in end points as to the composition of their interest area. The proximity values were more heavily penalized as the first dimension moved towards the limit of the interest area.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Presentity} & S_1 & S_2 & S_3 & S_4 & PS_1 \\
\hline
P_1 & 35 & 18 & 16 & 17 & - \\
P_2 & 25 & 23 & 10 & 15 & 0.684 \\
P_3 & 24 & 23 & 10 & 15 & 0.664 \\
P_4 & 23 & 23 & 10 & 15 & 0.644 \\
P_5 & 22 & 23 & 10 & 15 & 0.623 \\
P_6 & 21 & 23 & 10 & 15 & 0.602 \\
P_7 & 19 & 23 & 10 & 15 & 0.559 \\
P_8 & 18 & 23 & 10 & 15 & 0.537 \\
\hline
\end{array}
\]

Finally, I perform tests to verify that the size interest area or range being specified affects the outcomes of the calculations. Here all dimensions are kept constant but the range for the first dimension was progressively increased. The resulting effect being that a user can, in reality filter subsets of information by adjusting the range value or interest area to be more specific or generic. This puts total control of the groups clustered in an end point to the requirements of the end point itself and its applications and services residing locally.

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Presentity} & S_1 & S_2 & S_3 & S_4 & PS_1 \\
\hline
P_1 & 35 & 18 & 16 & 17 & - \\
P_2 & 35 & 18 & 16 & 17 & 1.000 \\
P_3 & 25 & 23 & 10 & 15 & 0.684 \\
P_4 & 25 & 23 & 10 & 15 & 0.687 \\
P_5 & 25 & 23 & 10 & 15 & 0.690 \\
P_6 & 25 & 23 & 10 & 15 & 0.693 \\
P_7 & 25 & 23 & 10 & 15 & 0.696 \\
\hline
\end{array}
\]
Deriving the context proximity and similarity over a distributed overlay, utilizing a publish/subscribe approach has the advantage of reduced computational operations in comparison to the less scalable centralized or brokering systems. In order to verify this, I consider the standard but simplified approach to finding and determining the context proximity for each presentity $P_i$ with respect to $P_1$. The task of calculating context proximity is divided into two distinct parts:

- Finding all instances of $P_i$ having similar context dimensions as $P_1$
- Calculating the context proximity of the set members

For an implementation using a context broker approach, such as IMS with presence profiles, the profiles for each $P_i$ must be requested. This is the least expensive query, making a single request for the profile, conducting a new search for each dimension would put an even greater strain on its ability to scale and perform over a large network of size $(N)$. Each return profile would be calculated with respect to dimensions, $S_i$ of $P_i$. Therefore cost of finding all instances of $P_i$ and calculating the relevance of each is stated as:

$$Cost^S = O(N) + O(N) + O(N) \quad (7.13)$$

for finding a single dimension, this is expanded to:

$$Cost^S = N + \sum_{v \in N} (2W_v) \quad (7.14)$$

where $W_v$ is the size of the presence profile returned in response to query. The size affects the cost of calculation as larger profiles require more computation for parsing, storage and retrieval.

The cost in Equation 7.14 is relatively greater compared to that which would obtained for the same results using the distributed overlay described in Section 7.4 Here I exploit two key properties: the ability to conduct searches with $0.5 \log N$ operations and the publish/subscribe component. Calculating the proximity for the same scenario is:

$$Cost^S = \sum_{u \in S} \left( O.5 \log(N) + \sum_{v \in RS} (2 \cdot W_v^{-1}) \right)_{u} \quad (7.15)$$

The resulting cost calculation derived in Equation 7.15 considers the properties of the supporting overlay. With its order preserving storage, a single query would find all the nodes with matching keys. The cost is therefore calculated as the cost of retrieving these values and performing a proximity calculation. The result set $RS$ is the set of
7.6 Evaluation

keys that are returned in response to the query and is doubled to account for the cost of retrieval and calculation. For calculation purposes: \( W_v \approx \text{number of context dimensions in a presence profile } P_i \).

With regards to updating such a group of entities, the cost is determined to be:

\[
Cost^R = RS + \sum_{v \in RS} (2W_v) \tag{7.16}
\]

Here, a broker is required to fetch the current values for all presentities in the result set \( RS \) and calculate the new proximity values. In my approach the improved cost is determined to be:

\[
Cost^R = \sum_{u \in R} \left( \sum_{v \in (dRS)} (2 \cdot W_v^{-1}) \right) \tag{7.17}
\]

By exploiting the publish/subscribe functionality of the overlay, no further queries are required, and the calculation cost is reduced to the cost of calculating the \( d \), the presentities with updated values. I simulated this calculation using a set of 1000 presentities and a total of 3 dimensions per presentity. The results are shown in Table (ref table) for the initial calculation and the updated calculation for both approaches. My approach shows improved performance with respects to the operational costs of determining all the presentity proximities with respects to \( P_1 \).

For a domain where the entire network size, \( N \) is 100, a user attempts to find a subset of other users based on his context. He searches on 3 dimensions and the resulting subset contained 300 other users within some proximity or him. Performing calculations on the computational effort required where I took into consideration the size of the data being exchanged and computed and the number of instances of each. Using a flat or brokering system to achieve this task in real time would required a total operational cost of 3000, while my distributed approach targeting smaller subsets and exploiting the organization on the overlay, would require a total operational cost of just under 400. The results were comparable for updating being more expensive in broker based systems than it was for my approach.

7.6 Evaluation

In this chapter I presented my approach to deriving a context proximity metric. This metric was derived in two parts, firstly there was the presentity similarity (CS) and the presentity distance \( PD \). The algorithm extended the general Euclidean distance to add normalization with respects to the range of the user. This contrasts to context maps created by Schmohl and Baumgarten [SB09] which normalizes to an arbitrarily chosen value and offers improvements with respects to the calculation of a user-centric
proximity. I present a single comparable value with a lower limit of 0.0 and an upper limit of 1.0 permitting applications and services to make assumptions and comparisons against other values I am therefore allowed to compare the behaviour of two entities and their relationships regardless of their context dimensions.

Additionally, these values are specific to the end points and its requirements and by adjusting them I am able to manage the size and affinity of the members of any subset. By providing the ability to specify weightings on each dimension, I offer an improvement over the approach used by Schmohl and Baumgarten and permit end points to create even more relevant subsets of context information. Improvements are offered over spatial only approaches such as Ambiesense or interaction dependent approaches such as the Smart-it friends project discussed in Section 5.3. I now provide a means of considering other dimensions of context when attempting to initiate activities based on user context proximity. Such derivation of context proximity are now more implicit and requiring minimal explicit user interaction. Approaches such as the Smart-it friends project could however compliment my proximity calculations by adding an additional context dimension.

With respects to scalability, I demonstrated though computational operations that the distributed approach is an improvement over centralized and brokering approaches such as IMS or SenseWeb. The computational costs have been shown to be reduced for making a standard proximity calculation and the effort required to keep it updated. My solution offers a new method for organizing relevant data in end points presenting smaller subsets of the global information set.

7.7 Summary

This chapter explored the main contribution with regards to the realization of an algorithm for determining context proximity. It further illustrated how such an algorithm can be realized on a distributed architecture, thereby benefiting from its inherent scalability. I detailed my contribution in the creation of an algorithm for estimating the context proximity among presentities. Such an algorithm permits the creation of schemata entities relevant to a user as one traverses the vast and dynamically connected things infrastructure. Here I addressed the problem of being able to present relevant subsets of global sensor information to applications and services residing in end-points. I have shown that a presentity distance and proximity metric can be obtained with regards to defining context proximity. Additionally, I demonstrated though a simulation of such a derivation, that these metrics consider the requirements of the services residing in the end points by specifying the range of interest and the dimensions to be considered. Applications and services can further specify the weighting for each dimension in order to achieve the most optimized subset of information. By extending an existing gossiping algorithm I have shown that this can be implemented relative to a distributed overlay providing improvements with respects to the computational cost of deriving proximity metrics. The results as shown, promises improvements when compared to broker systems even with the same algorithm implemented.
Chapter 8

Estimating Sensor Ranking

Section 3.4 motivated, by means of a discussion over current theories and related work, the need to create solutions that enable the ranking of sensors over heterogeneous, distributed context networks. Existing solutions being incapable of enabling the support required, both in a distributed and centralized approach. Other approaches are fixated on ranking around the information rather than the information source itself.

A quality negated by the fluidity of context networks. This chapter presents one approach to calculating sensor ranking (See Paper X) based on their general usage patterns. Such a sensor reputation algorithm is further shown to be implementable over a distributed architecture without the need for any centralized coordination. Here I further address the problem of being able to create relevant and accurate subsets of information relevant to the realization of an application and service. By presenting metrics indicative of a sensor’s reputation, subsets can be created that consider more reputable sensors. By deriving a metrics for comparing sensor ranking and reputation in a distributed architecture, these metrics can be used to add value to the sensors being used to derive context information.

8.1 Overview

The schema objects discussed in Section 6.3 permit presentities to create collection of sensors, contributing and expressing context over a presentity. Using this approach creates a local awareness of all the instances where a sensor \( s \) has been utilized by \( P \) in schema construction. All the schemata that have been used by \( P \) is also known and can therefore be used to derive some representation of the relevance of \( s \) relative to \( P \). Such a value, it can be reasoned, represents the localized ranking of \( s \). The local node at which \( s \) resides should therefore able to collect and aggregate these values, indicating an overall, near global, ranking value of \( s \).

The ranking algorithm consists of two main components: a local ranking value and a global ranking value.
8.2 Localized Ranking

Firstly, the local ranking value for $s$ must be determined with respect to $P_i$, and then aggregated into a global ranking value for $s$. The approach involves an adaptation of the Term Frequency - Inverse Document Frequency algorithm [Rob04]. This algorithm, initially used to calculate the importance of a query term with respect to document corpus, provided a simple but representative metric for ranking documents with respect to a search query.

The algorithm shown in Equation 8.1 is modified with respect to sensor $s$, schema $r$ and presentity $P$. A sensor in a schema is considered here to be analogous to the query term in a document and is expressed as follows:

$$SR_{s_i}(P) = \log \left( \frac{|R|}{\{r : s_i \in r\}} \right) \times \left( \frac{t}{T} \right)$$  \hspace{1cm} (8.1)

$SR = \text{sensor ranking}$

$R = \text{corpus; total collection of schemata at } P$

$s = \text{a sensor}$

$r = \text{all schemata containing } s$

$T = \text{total time over all schemata in } R$

$t = \text{total time over all schemata containing } s$

This provides for a representative metric as to the importance of sensor, $s$ relative to $P$. Further consideration is given to scenarios where some presentities will be less dynamic or mobile with respect to $s$. Such an example might be a sensor located in a store; the employees working in the store will by default almost always utilize the sensors that are local to the store accounting for a disproportionately higher value for:

$$\log \left( \frac{|R|}{\{r : s_i \in r\}} \right)$$  \hspace{1cm} (8.2)

In such scenarios, all stores within a shopping area would have high values, granted solely by the employees themselves.

Equation 8.3 considers more dynamic presentities traversing an Internet of Things. A person that travels around the city more often interacts and creates more context schemas in fulfillment of service delivery. This is represented by larger ratio of $R$ to $\{r : s_i \in r\}$. Such presentities, it can be argued, are more representative of the ranking that should be associated with $s$, relative to the wider sensor ecosystem.
Another scenario being that we penalize malicious nodes may that might attempt to collude or independently attempt to inflate their rankings by creating a disproportionate number of schemas. In such scenarios, the result of Equation 8.2 would move closer to a value of 1. By taking its logarithm, instead, we adjust this such that for \(|R| = \{r : s_i \in r\} = 1, SR = 0\). Thereby having a very small effect on the sensor ranking.

A sensor that has very few schemata with a presentity and used for a longer period of time relative to the other schemata would be accorded a lower ranking value to compensate for such a scenario. Similarly, a sensor that is used a lot and for extended periods, would have a relatively smaller value for its ranking. Ideally, a sensor that is used a lot but for short periods are more indicative of a person interacting within a digital ecosystem and being more representative of the affinity to a sensor. Another scenario being that we penalize malicious nodes may that attempt to collude or independently attempt to inflate their ranking disproportionately creating and using a number of schemas. In such scenarios, the result of equation 8.2 would move closer to a value of 1. By taking its logarithm, instead, this is adjusted such that \(SR \rightarrow 0\), thereby having a very small effect on the sensor ranking.

### 8.3 Time Limited Localization

While (8.1) permits the calculation of ranking over the entire interactions of \(P\), the need arises to be able to calculate sensor rankings at some given point \(v\). Such a scenario would be useful when trying to rank sensors that are in use at an event or situation occurring in a localized area. Here, the sensors could be tasked with reporting ranking limited by some time duration such that:

\[
SR_{si}^{(P)} = \log\left(\frac{\{r : s_i \in r\}}{|R_v|}\right) \times \left(\frac{t_v}{T_v}\right) 
\]

\(v = \text{time duration}\)

Each presentity now has a value for its ranking of \(s\), both historically or over some period. This metric which permits the presentity to evaluate the usefulness of \(s\), if it encounters it again or if \(s\) is amongst several other sensors.

### 8.4 Global Ranking

The second component of my approach is a global aggregation of all the local ranking values \(SR\) assigned to \(s\). This Global Ranking \(GR\) is calculated by finding the sum of all \(SR\) of \(s\) such that:
Algorithm 2: Ranking Sensors and Information Points

loop

{at the local node, $i$}

determine the size of the local corpus $R_i$ of schemata

for all information points $s$ attached to $P_i$ do

determine the ranking value with respect to $R_i$
assign this as the local ranking value $SR_i$
forward this to the global domain owner $D$
end for

{at each domain owner, $j$}

for all information points $s_i$ residing at $D$ do
aggregate the all values received for $SR_{s_i}$
calculate the new value for the domain rank $DR_j$
end for

end loop

\[ GR_s = \sum_{v \in P^s} R_v \] (8.5)

The value for $GR$ value is continually calculated as new schemas are created with reference to $s$. Presentities, by means of the publish/subscribe approach sends updated information to the owner of $s$. Only the delta of the size of $R$ and $r$ are needed in order to perform a recalculation. Therefore if a presentity does not change its degree of relationship with $s$ then no calculation is redone.

The resulting values derived above, can be used as indicators or relevancy or importance of sensors in an Internet of Things.

8.5 Approach on a Distributed Architecture

Within a distributed architecture, the implementation of such a sensor ranking approach gains the best implementation with respect to performance and its ability to scale. A presentity would likely not reside on the same node as the sensor that it is trying to use.

A distributed solution, implemented on architecture detailed in Section 5 for messaging support, would implement the distributed ranking algorithm described in this chapter. A node using a sensor in a schema in order to represent the context information for a presentity, calculates the localized ranking value and forwards it to the node responsible for the sensor. In DCXP, this would be the node that registered the sensors. The node owning the schema, aggregates the sensor ranking and makes this available on the overlay to any other interested nodes. The benefit of this approach is
that sensors calculate values locally with no centralization needed, deriving its scalability properties from the underlying DCXP infrastructure. Such an algorithm is summarized in Algorithm 2.

8.6 Verification

Deriving a value representative of the ranking of a sensor over a distributed overlay, utilizing a publish/subscribe approach has the advantage of reduced computational operations in comparison to the less scalable centralized or brokering systems. In verification of this I consider the cost in terms of the amount of effort that would be required to maintain a ranking approach both in a broker or centralized implementation and relative to the distributed implementation discussed earlier in this chapter.

I consider the cost of creating a searchable ranking index for a sensor $S_1$, being used by several presentities. In determining the ranking of within a centralized or broker-based architecture, a search engine server would be required to crawl all presentities, fetching profiles and aggregating the information locally. It would then proceed to calculate the ranking for each sensor and share this information with other users wishing to access it. The cost of doing this in its simplest form is expressed as:

$$GR(S_i) = 3N$$

(8.6)

While for my distributed approach the cost would be:

$$GR(S_i) = \sum_{v \in RS} 2v$$

(8.7)

Where $RS$ is the result set, $N$ is the network size, and $v$ is each presentity using $S_1$.

Consider a network of size 1000 presentities, 300 presentities are using the sensor $S_1$. The cost using a centralized or broker based approach would be 3000 while for my distributed approach this would fall to 300. This as I would not need to scan the entire network for instances of presentities using $S_1$ but rather simply relied on the subscription requests to derive a value for $GR(S_1)$. Further, updating the ranking value would have a similar cost for the centralized approach as it would be required to do a new scan each time to fetch current usage values. The distributed approach, however would simply update its values from new subscription requests.

8.7 Evaluation

In this chapter I introduced an algorithm for calculating sensor ranking based on usage patterns in context awareness implementations. The resulting costs with respects
Estimating Sensor Ranking
to computational overheads shows and improvement over centralized approaches and this permitting, the derivation of such a metric within reasonable computational constraints. The algorithm has improvements over a priori approach such as Google indexing by being able to determine a ranking at run time and when requested by a user or end point.

While this is an improvement over the performance of what would be expected ion a centralized approach, the algorithm could consider additional dynamics influencing ranking values such as the reputation of the presentities deriving these values.

8.8 Summary

In this chapter I introduced a algorithm for deriving a sensor ranking value within a distributed architecture. This algorithm considers the usage patterns of presentities with regards to their interaction use of sensors in order to derive or define context schema. In contrasts to a priori based approaches such as web ranking, I calculate sensor ranking values in real time and in direct response to the current usage. Additionally, I detailed how such an algorithm can be implemented on a distributed architecture utilizing a publish subscribe approach to determining current usage negating the need for expensive network crawling and indexing. This algorithm however could be further improved to consider additional dynamic such as presentity reputation and domain reputation.
Chapter 9

Conclusions and Future Work

The gradual shift towards an *Everything, Everywhere* computing paradigm is mandating a parallel but equally critical shift in the way applications and services are provisioned. As computing becomes increasingly ubiquitous, users are demanding that such applications and services are capable of being provisioned in response to their situation; the concept of *Context Aware Computing*. Such a shift in the way services are provisioned is underpinned by the ability to derive an understanding of the vast amounts of information possessed by the growing digital ecosystem being interwoven into our very social fabric.

The need to understand and manipulate user context information has driven research into systems and methodologies for realizing more intelligent interaction among social artefacts. Solutions towards context information provisioning and modelling have often fallen short with regards to their ability to scale well within a truly large-scale pervasive environment. Their limitations are also emphasized with regards to the provisioning of useful metrics permitting a common understanding of the states, relationships and interactions among entities in an interconnected *Internet of Things*.

In response to this, I detailed my contributions towards a distributed context modelling solution for context aware applications and services. I detailed how the information obtained in such a model may be used to derive useful and representative metrics enabling a greater understanding of the states, relationships and interactions among entities. This, by being able to organize relevant and useful subsets of global information in the end-points where applications and services resides. Such dynamic self-organizing structures indicative of the interactions among real-world artefacts. Additionally I described how my approaches can be implemented on a a distributed overlay realizing scalability and reducing computational overhead.

**Distributed Models Supporting Context Dependent Applications and Services**

I detailed an overview of the CII model and detailed its subsequent extension, the DCII model. The CII model provides an expressive and dynamic approach to modelling
context information. As an object-oriented ontology it permits the addition of new concepts and relationships. In extending this model, I detailed how this is achievable relative to the distributed overlay detailed in Chapter 5. This solution creates the support required to implement the real-time dependent context services conducive with an Internet of Things. It addresses the issues surrounding the scalability of centralized sensor information access by leveraging DCXP maintaining the proven real-time information dissemination unachievable with previous approaches such as Senseweb.

This distribution premises the ability to build and modify complex sensor relationships in real-time to reflect the dynamic realities of the Internet of Things. It also described the introduction and use of the sensor schema to model the evolving context information related to presentities. This is maintained through a publish/subscribe interface. Additionally I introduce the concept an an object schema as a means of organizing and maintaining collections of related objects in end points over the distributed architecture. With this I can create sets of context information related to an entity at end-points within the architecture. By taking the approach of schema evolution, I am able to obtain information on the state and behaviour on context information subsets. The result is a solution with which to derive relevant metrics needed to evolve and maintain complex networks of context information.

An Algorithm for Estimating Context Proximity

The first of two types metrics I introduced was that of context proximity. Within a future Internet of Things, users will require useful metrics in order to understand the digital ecosystem in which they are embedded. With regards to the derivation of useful metrics, I introduced an approach to measuring context proximity. It described an algorithm for creating clusters of related entities determined by each member’s distance relative to an ideal value, the current context value for a presentity. This algorithm provides for a means of exploring a context network, consequently discovering other presentities that possess a context value within range and by extension indicating a context within some degree of similarity.

The algorithm determines the Jaccard Coefficient or in this case the schema similarity, based on the common groups of sensors between presentities. This similarity metric is then adjusted using the value of degree of similarity over the context values. Applications and services can further specify the weighting for each dimension in order to achieve the most optimized subset of information. This permits for a determination of significant presentities within a given group of connected people. This metric, the Context Proximity, proxies a new avenue for deriving new sets of sensors and presentities with which to connect and derive new representations of context. I demonstrated though a simulation of such a derivation, that these metrics consider the requirements of the services residing in the end points by specifying the range of interest and the dimensions to be considered.

The application of an existing gossiping protocol over the DCXP architecture creates rules for forwarding and grouping sensors occupying a context values within a given a pre-determined range. Both protocols and architecture have demonstrated
the ability to perform and scale well providing improvements with respects to the computational cost of deriving proximity metrics. The results as shown, promises improvements when compared to broker systems even with the same algorithm implemented and thus an adequate response to the problem of being able to create, from global context information, relevant subsets of information in end-points.

**An Algorithm for Estimating Sensor Ranking**

In addition to determining the proximity between interested presentities, navigators of a future Internet of Things will require further metrics that construct an overview of a sensor’s ranking, reputation or usefulness. I detailed the adaptation of an existing document ranking algorithm towards deriving such a metric. In contrast to solutions that rank sensors within the confines of wireless sensor networks, such a solution seeks to provide a metric representative of an overall global ranking; elevated from localized wireless networks and in response to the usage patterns of context-centric infrastructures.

I further described how such a solution can be realized relative to a distributed connected things approach realizing improvements over centralized and broker based infrastructures with respects to the operational costs of calculating and updating a ranking value.

**An Architecture for the Provisioning of Sensor Information**

In this thesis, I presented support for the distributed provisioning of sensor information. Such a support is capable of handling and mediating the frequent and varying updates that can arise from ah hoc sensors deployed in the wild. This was achieved through the use of the DCXP protocol realized on a distributed P-Grid peer-to-peer overlay. The solution is in response to the requirements raised in Section 3.1 with regards to a solution comprising of a scalable responsive overlay, an interoperable exchange protocol and a flexible and open naming scheme.

I introduced P-Grid as the overlay of choice along with the use of object-oriented databases for local persistence in an effort to improve performance and natively persisting context objects. I also added the **TRANSFER** and **SET** primitives as well as a distributed Context Storage on the overlay and using UCI’s for naming objects outside of raw sensor values. These objects I also persists alongside sensor information for fast look-ups and name resolution.

The solution provides for true self organization and is capable of handling the varying levels of connectivity inherent in solutions dependent on internet of 3G radio networks. The realization of a distributed persistence enabled the storage and querying of context information among applications and services. Such information has an increased level guaranteed freshness by virtue of the improved real-time performances in contrast to more centralized solutions. Maintaining interoperability with existing presence systems such as IMS, enables sensor information provisioning in mobile applications to a vast, existing and increasingly growing population of mobile phones,
which may not be participants in a DCXP infrastructure.

**Future Work**

The material presented in this thesis aptly lends itself to future extensions and improvements. The provisioning of sensor information could benefit from further investigation into the incorporation of mobile devices without the need for the mobile proxy element. The creation of a more robust persistence layer could enable more expressive querying of sensor information. The distributed object model could be realized directly on top of this persistence layer with little or no need for localized persistence, instead creating a single universal connected model. With regards to the derivation of metric, the proximity and ranking algorithm could be extended to accommodate more factors such as presentity reputations, composite values for historical and temporal ranking data and a unification of both algorithms to create a single set of metric supporting dynamic the relationships and interactions inherent in a future *Internet of Things*. Additionally the self-organization properties of the overlay can be exploited to create real time clusters of presentities organized around the key-space and derived using these metrics relative to a general application requirement. These metrics and the use of clustering approaches can be incorporated into a distributed recommendation solution enabling grouping and recommendation in response to context information.
Bibliography


