Självständigt arbete på avancerad nivå

Independent degree project – second cycle

Datateknik 30 hp
Computer Engineering 30 credits

A Move Towards Practicability
Context Information and User Profile Management for the SensibleThings Platform

Sileshi Chukala
Abstract

Achievements in academia and industry indicate a move towards the creation of a giant smart system including objects of different sizes, shapes, and purposes where the Internet will be so diverse and more resourceful with the incorporation of different smart devices. Mid Sweden University took a first step towards providing an appropriate platform for Internet of Things applications known as SensibleThings for the emerging intercommunication of “objects”, called Internet of Things. The SensibleThings platform has produced components for sharing sensor-based context information on the Internet, in order to enable Internet of Things applications. In this thesis the plan was to contribute to filling the gap between the IoT and developers in academia as well as in the industries. The aim of the project was to create basic functionality to enable users to find new interesting users on the IoT and prioritize existing users. To this end context information and user profile management is proposed and presented. Furthermore, a comparison between the platform with the proposed extension as well as without was compared and examined.

Keywords: User profile, Context profile, IoT platform, SensibleThings.
Acknowledgements

I would like to thank friends and family who proved their love in times of social and economic stress during the course of this project. Most of all I would like to thank God for helping me endure the bad times. In addition, I would like to express my warmest gratitude to Professor TingTing Zhang for her time and review of my work as well as the adjustments she suggested. I would also like to thank supervisor Stefan Forsström for his valuable comments and guidance.
Table of Contents

Abstract ........................................................................................................................................ ii
Acknowledgements ...................................................................................................................... ii
Table of Contents ........................................................................................................................... iii
Terminology ...................................................................................................................................... v

1 Introduction ..................................................................................................................................1
1.1 Background and problem statement .......................................................................................1
1.2 Overall aim ...............................................................................................................................2
1.3 Concrete and verifiable goals ....................................................................................................2
1.4 Scope .......................................................................................................................................2
1.5 Outline .....................................................................................................................................3

2 Theory .........................................................................................................................................4
2.1 The Internet-of-Things ..............................................................................................................4
2.1.1 Internet of Things challenges and issues ...........................................................................7
2.1.2 Internet of Things Platform ............................................................................................10
2.1.3 Parameters of IoT platform ..............................................................................................11
2.2 Pervasive and ubiquitous computing ......................................................................................13
2.3 Context .....................................................................................................................................14
2.3.1 Context categories ............................................................................................................14
2.4 Context awareness ...................................................................................................................15
2.4.1 Features of context-aware applications ...........................................................................16
2.5 Peer-to-peer computing ..........................................................................................................18
2.6 Group communication management mechanisms ....................................................................19
2.7 The SensibleThings IoT Platform ..........................................................................................21
2.7.1 The SensibleThings Platform architecture .......................................................................21
2.8 Distributed context exchange protocol (DCXP) .....................................................................23
2.8.1 DXCP naming scheme ......................................................................................................24
2.8.2 DCXP messages ...............................................................................................................25
2.9 Related work ............................................................................................................................26
2.9.1 WC3 CC/PP standard .......................................................................................................28
2.10 Proposed user profiling ..........................................................................................................28

3 Methodology ...............................................................................................................................30
3.1 Building a knowledge base ....................................................................................................31
3.2 Context information and user profile management ...............................................................31
3.3 Proof-of-concept application .................................................................................................31


<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>Theoretical evaluation and performance test</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>Design and implementation</td>
<td>34</td>
</tr>
<tr>
<td>4.1</td>
<td>The proposed extension</td>
<td>35</td>
</tr>
<tr>
<td>4.1.1</td>
<td>User-profile management</td>
<td>35</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Context-profile management</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Proof-of-concept application</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>Design summarization</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>Results</td>
<td>44</td>
</tr>
<tr>
<td>5.1</td>
<td>The proposed extensions</td>
<td>44</td>
</tr>
<tr>
<td>5.2</td>
<td>The proof-of-concept application</td>
<td>46</td>
</tr>
<tr>
<td>5.3</td>
<td>Usability test and evaluations</td>
<td>48</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Survey parameters</td>
<td>48</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Data collection approach</td>
<td>49</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Technical comparison</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>51</td>
</tr>
<tr>
<td>6.1</td>
<td>Challenges and coding errors</td>
<td>52</td>
</tr>
<tr>
<td>6.2</td>
<td>Discussion</td>
<td>52</td>
</tr>
<tr>
<td>6.3</td>
<td>Contribution and impact</td>
<td>53</td>
</tr>
<tr>
<td>6.4</td>
<td>Ethical considerations</td>
<td>53</td>
</tr>
<tr>
<td>6.5</td>
<td>Future work</td>
<td>53</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>55</td>
</tr>
<tr>
<td>Appendix A</td>
<td></td>
<td>61</td>
</tr>
<tr>
<td>Questions I</td>
<td></td>
<td>61</td>
</tr>
</tbody>
</table>
Terminology

Abbreviations and Acronyms

GPS  Global Positioning System
IOT  Internet-of-Things
IBSG Internet Business Solution Group
VCR  Video Caste Record
PDA  Personal Digital Assistant
DHT  Dynamic Hash Table
JXTA Juxtapose
CAN  Content Addressable Network
IP Internet Protocol
TCP Transmission Control Protocol
QoS Quality of Service
QoE Quality of Experience
DHCP Dynamic Host Configuration Protocol
Wi-Fi Wireless Fidelity
UCI Universal Context Identifier
UDP User Datagram Protocol
RUDP Reliable User Datagram Protocol
P2P Peer-to-Peer
API Application Programming Interface
DCXP Distributed Context Exchange Protocol
PA Profile Aggregator
RDF Resource Description Framework
XML Extended Markup Language
XSL Extensible Style Sheet Language
SQL Structured Query Language
RDQL RDF Data Query Language
URL Uniform Resource Locator
OWL Ontology Web Language
XDM XML Document Management
XDI eXtensible Data Interchange
CC/PP Composite Capabilities/Preference Profile
W3C World Wide Web Consortium
POC Proof of Concept
WHO World Health Organization
DCXP Distributed Context Exchange Protocol
NAT Network Address Translation
1 Introduction

The Internet is in the process of forming a nebula of objects (both smart and obscure) in its core. The emerging high-tech and cheap computational/communication devices plus the increasingly cheap sensors are creating an environment where the Internet will be playing a central role in modern human social activity. Achievements in academia and industry indicate a move towards the creation of a giant smart system including objects of different sizes, shapes, and purposes where the Internet will be so diverse and more resourceful with the incorporation of different smart devices. It is believed that by year 2020 the Internet will connect 26 billion devices [1]. These interconnections are known as Internet of Things. Compared to today’s Internet, the devices in the Internet of Things go far beyond servers and computers, which were the endpoints connected to the Internet back in the early days.

Mid Sweden University took a first step towards providing an appropriate platform for the emerging intercommunication of “objects”, called Internet-of-Things or IoT. The current version of the platform is known as SensibleThings [2] forked from the previous project [18]. The research team at the Department of Computer and Systems Sciences has already implemented the basic platform ready for Internet-of-Things developers around the world.

1.1 Background and problem statement

The SensibleThings platform has produced components for sharing sensor based context information on the Internet, in order to enable Internet-of-Things applications. Sensor information should originate from numerous different sources, such as sensors attached to people, houses, vehicles, mobile phones, or sensors out in nature connected to sensory networks. The current SensibleThings platform demands a new functionality to structure interacting peers in the platform.
Therefore, in this thesis the plan is to contribute to filling the gap between the IoT and developers in academia as well as in the industries. More specifically, an extension for context information and user profile management is developed.

1.2 Overall aim

The overall aim of the project is to create basic functionality enabling users to find new interesting users on the IoT and prioritize existing users. To this end context information and user profile management is proposed. The proposed solution is expected to fill the breach in the particular area pinpointed. Furthermore, an application which is able to consume this extension is designed and implemented to prove that the concepts in the proposed solution are practical.

1.3 Concrete and verifiable goals

The goals of this thesis have been to implement context profiles and user profiles in the SensibleThings platform. More specifically, this thesis will include:

1. A Literature review: To build the solid knowledge base needed to successfully carry out the aim of this Master’s thesis.
2. Design and develop functionality for context information and user profile management for the IoT Platform.
3. Develop an IoT proof-of-concept application that utilizes the developed functionality for context information and user profile management extensions and runs on the top of SensibleThings Platform.
4. Evaluate the project outcome and illustrate the overall gain of the solution proposed.

1.4 Scope

This thesis focuses on proposing a context information and user profile management solution for Internet-of-Things platforms. More specifically, the proposed solution is targeted to fit the IoT platform from MIUN, SensibleThings Platform [2]. Therefore, implementation of the solution as an extension to the existing Library falls within the scope.
Out of the scope of this Master’s thesis is extended evaluation of the deliverables. A usability test has been carried out on the proposed outcome of the context information profiling and user profiling, however, it was done to match the project time and budget limitations.

1.5 Outline

This thesis is organized into six chapters: Chapter 1 gives the introduction, background and problem motivations of the thesis. Chapter 2 describes the theoretical framework and gives an overview of related work. Chapter 3 describes the general methods used to achieve each specific goal and the methods used to implement the prototype. Chapter 4 describes the overall design issues and design choice made for the extension, implementation of the overall functionality of the add-in and concept-proof-application. Chapter 5 presents the results and testing to implement as well as the results achieved and the performance test of the resulting developed extensions. Chapter 6 presents the conclusion drawn from this thesis work and provides ideas for future developers in terms of work on context profiles.
2 Theory

This section presents the background and related work. It provides an introduction to the Internet-of-Things, pervasive and ubiquitous computing, context information, peer-to-peer computing, group management, the SensibleThings platform, user profile management, and related work.

2.1 The Internet-of-Things

The term Internet-of-Things was first coined by the British technology pioneer Kevin Ashton. He used the term as the title of a presentation at Procter and Gamble (P&G) in 1999 at Auto-ID Labs [41]. The expression itself and the concept of Internet of Things has been defined by different scholars in different ways. The Internet-of-Things (aka IoT) [1], is a communication network connecting things that have naming, sensing, processing, communicating and information sharing capabilities. Devices like computers, smartphones, home appliances, home gateways, refrigerators, washing machines, watches, TV, cars, sensors and actuators are connected via IP addresses to collect, measure, analyse and manipulate the physical properties of an object [2], but all the interconnected things are not exclusively used for electronic devices, they also include any item that is recognizable, readable, locatable, addressable and controllable via the Internet [3]. The cooperation and integration of all these real-world things, provides anything, anytime, anyway, anyhow, anywhere, connectivity [42].
According to the Cisco Internet Business Solutions Group (IBSG) [4] research study, there was about 12.5 billion devices connected via the Internet in 2010, and they predicted the total number of connected devices to reach around 50 billion in 2020, if the number of smart devices per person increased. However, it is currently believed that by the 2020, the Internet will connect 26 billion devices [1]. The Internet dependency on people to supply information will change to smart things and ubiquitous sensors. These devices will be able to generate the input data themselves. This means data is collected and analyzed in real time. Thus, from the interconnection of the things we get more personalized, automatized and intelligent context information from the real world in real time through the use of a variety of Internet of Things applications or machine-to-machine applications [3].

IoT applications cover all areas of human need some of these areas as well as relevant major scenarios are listed and summarized in Figure 2 below.
There are a lot of recent IoT products released on the market, for example, the NEST Thermostat [44], which is produced with the aim of saving energy in the home environment, the WEMO Remote [45], but unlike to the NEST Thermostat it is used for home automation, the Ninja Sphere Home controller hardware [46] is designed to bridge all home smart devices, the Intel Quark Processors [47] is a new processor released by Intel for Internet-of-Things to provide flexible, low-cost connectivity, integration, compatibility, end-to-end security, manageability, performance enhancement and also low-power computing for a wide variety of applications, the Tractive GPS Pet Tracker device [48] is a real-time tracker used to track the position of pets and other animals, the Corventis Wireless Cardiac Monitor device [49] is an eHealth monitoring device continuously monitoring the heart activity of a patient relative to their cardiac rhythm by electrocardiography; these are some recent example of IoT products.
2.1.1 **Internet of Things challenges and issues**

Some of the biggest challenges and issues in the development of Internet of Things is a lack of uniform global technical standards, lack of security and privacy, bandwidth, connectivity, sensor network management, energy, data management, analysis, visualization, storage, and so on [50]. But standardization, security and privacy are the most critical issues that should be given greater consideration; there is an urgent need for a solution to overcome these challenges and problems.
Internet-of-Things is a global concept that bridges the virtual world and the physical world by allowing them to connect all things to the Internet all around the world [51]. However, the idea of connecting anything to the Internet is a major challenge for standardization, but the realization of the Internet-of-Things could be achieved by developing global interoperable standards [51].

Global standards offer reliability, interoperability and compatibility of things that are connected on a global scale. By estimating is future business opportunities there are multiple, regional, international standards organizations as well as several forums and consortia that have contributed to the IoT standards.

Some of the international standard development organizations working individually and in collaboration relevant for the Internet-of-Things include ITU (International Telecommunication Union), ISO (International Standard organization), IETF (International Engineering Task Force), IEC (International Electrotechnical commission), IEEE (Institute of Electrical and Electronics Engineers), and regional organizations such as IERC (The European Research Cluster), Global Standards Collaboration on M2M Standardization Task Force (MSTF), and forums and consortia such as 3GPP (Third Generation Partnership Project), oneM2M, Near Field Communication (NFC), Open Geospatial Consortium (OGC), IPv6 Forum, and IoT Forum are all working on establishing standards for IoT.

According to the oneM2M website “oneM2M consists of eight of the world’s most excellent ICT standards development organizations such as, ARIB (Association of Radio Industries and Businesses (Japan)), ATIS (Alliance for Telecommunications Industry Solution(North America)), CCSA(China Communications Standards Association (China)), ETSI (European Telecommunications Standards Institute (Europe)), TIA(Telecommunication Industry Association(North America)), TSU (Telecommunication Standards Development Society
Development Society (India), TTA (Telecommunication Standards Development Society (Korea)), and TTC (Telecommunication Technology Committee (Japan)), together with seven industry fora, consortia or standards bodies (Broadband Forum, CEN (European Committee for Standardization), CENELEC (European Committee for Electrotechnical Standardization)), GlobalPlatform, HGI (Home Gateway Initiative), Next Generation M2M Consortium, OMA (Open Mobile Alliance) have been working in collaboration on the standards for M2M and the Internet of Things” [56].

**ISO/IEC Standard**

The main focus of ISO/IEC IoT WG is to develop standards for sensor networks, terminology, taxonomy, reference architectures, guidelines for interoperability and specific aspects of sensor networks [51].

**IETF Standard**

The motto of the IETF IoT working group is “a true Internet of Things requires ‘things’ to be able to use Internet protocol”. Constrained devices or things are able to use an optimized version of Internet protocol to connect with the Internet [52]. The main focus of the group has been to develop communication protocol suits and standards for the future Internet and constrained devices such as IPv6, 6LoWPAN (IPv6 low power WPAN), defining IPv6 header compression and adaptation layer suitable for low power constrained radio links, RPL (IPv6 Routing Protocol for low-power and Lossy Networks) for routing, CoAP (Constrained Application Protocol), a web transfer protocol for constrained devices and networks that runs on any device that supports UDP and UDP analogue, and Transport Layer Security/Datagram Transport Layer Security (DTLS) profiles for constrained IoT devices [52].
IEEE Standard

IEEE Standardization activities mainly focus on the low layer protocols known as the physical layer (PHY) and the MAC layer. IEEE has been a stepping stone for the early foundation for the IoT with the ZigBee (802.15.4) standard for low power short range radio bands, used specifically in industrial, scientific, and medical (ISM) bands, smart grid metering and industrial automation [53].

The enhanced standard ZigBee, 802.15.4g PHY, allows larger packet transmission and fits with the IPv6 minimum value, and 802.15.4e MAC, allows deterministic properties with the Time Slotted Channel Hopping (TSCH) mode of operation [53].

2.1.2 Internet of Things Platform

According to the recent Gartner identification of the top ten IoT Technologies that will be needed in 2017 and 2018, the IoT platform is one of the top ten [54].

IoT platforms are one of the backbones of the realization of IoT, and it is vital to enable connectivity between systems, objects and people in general IoT platforms are required to develop and manage IoT applications and services. An IoT platform can also be called IoT middleware because of its functional role as an intermediary between hardware and application layer to overcome the heterogeneity problem throughout the IoT system.
2.1.3 Parameters of IoT platform

Currently, there are various IoT platforms available on the market that can be used to develop IoT solutions. Some of the most popular IoT platforms that are widely used to develop and provide IoT solutions are Amazon Web Service (AWS) IoT, Microsoft Azure IoT, ThingWorx IoT, IBM BlueMix, ThingSpeak, Digital Service Cloud, Google Cloud Platform, Axeda and Oracle Java Embedded.

Some of the most popular platforms in IoT group their architecture, either according to platform-as-a-service (PaaS), Software as a Service (SaaS) or Infrastructure as a Service (IaaS). However, there is no stated global standard set by a global standard organization. Thus, the choice of appropriate platform completely depends on the IoT solution providers. Therefore, the key features of choosing the appropriate IoT platform for an IoT application is important for IoT solution providers because the available platform is unable to provide all application requirements, so the choice of appropriate platform is critical for enterprises.

Recently an online survey based on the results of the Google search engine was conducted by Solari [57] to identify how to choose and evaluate IoT platforms as a result they identified six key features for the
choice of IoT platform for enterprises such as completeness, performance, security, flexibility, communication and business focus [57].

**Completeness of an IoT system**

The IoT platform should be interoperable and integrate by supporting heterogeneous device connectivity, like sensor and actuator technologies, and also include device monitoring, maintenance, data management, analytics, and integration with other external interfaces.

**Performance**

An IoT platform should be able to provide the option of big data management and data traffic optimization to ensure low latency and good customer experience.

**Security**

Security is important for all information technology solutions, specifically in IoT. Privacy and security must be a major concern throughout all levels, for example device level security, infrastructure level security, data level security, storage security and applications security.

**Scalability**

The scalability or flexibility of the IoT platform depends on the underlying architecture in which the platform is developed, and scalability should not only relate to the ability to accommodate a larger number of connected things, but also the ability of adaptation to new protocols, additions and adaptation of new features.

**Openness**
Openness is one of the features of the IoT platform and it relates to the communication between the IoT platforms, devices, applications and the interoperability of different platforms.

**Business focus**

When building an IoT application that provides a valued service to society or a company, choosing the proper platform is an important task. The choice must be based on cost for the individual user and the company cost.

### 2.2 Pervasive and ubiquitous computing

Pervasive (ubiquities) computing is an area that emerged as a result of the success of embedded system. The term ubiquitous computing was coined by Mark Weiser in 1988 [5], and the basic concept was that computing and processing are embedded into everyday devices that are accessible everywhere in the environment.

These everyday accessible devices include any intelligent device capable of communicating and interacting that is context-aware, and exchanges information everywhere, i.e. beyond the usual desktop and server machines. Hence, ubiquitous means constantly available, omnipresent or present everywhere at the same time, and pervasive, which is the capability of computing and communicating, it includes distributed and mobile computing.

Nowadays there is a large number of available technologies used to enable the notion of ubiquitous computing, among these enabling technologies. Some of them are a development of processing capabilities such as cheaper, smaller, and more energy efficient, the increase of storage capacity in terms of size and speed, the availability of high bandwidth, low latencies, and ad-hoc networking, the accessibility of sensors of different types, speed, accuracy, price, and robustness, the spreading of dedicated devices such as smart mobile phones, digital cameras, VCR, and PDA, and actuators. Therefore, ubiquitous and
pervasive computing will enable various wireless Internet-of-Things applications in different real world scenarios.

2.3 **Context**

The term context has different meanings in different perspectives and scenarios, but in order to be used effectively for applications, we need a better understanding of what the context is and how it can be used in interactive applications. It needs a general and common operational definition this mainly helps application developers to choose what context to use in their application. Context is defined differently in different scenarios, for example synonyms are background, circumstance, situation, framework, perspective, environment, etc. of an entity. In an online dictionary context is defined as “the set of circumstances or facts that surround a particular event, situation, etc” [59]. The problem is how used in to practice. An operational definition is needed to encompass all context information. [Definition and Implementation of Context Information [6]], [Understanding and Using Context [8]], [Context-Aware Group Communication in Mobile Ad-Hoc Networks [9]], [A Mobile Human-computer Interaction Perspective on Mobile Learning [10]] provided an operational definition of context, but the most widely accepted and used definition of context in the literature is the definition provided by Dey and Abowed [7].

*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.*

2.3.1 **Context categories**

The categorization of each piece of context information in different context types is essential for application developers to obtain refined context information for their applications. Unlike to the context definition different papers suggested different forms of context type categories. In [Mobile Data Management], [Chen] [20] suggests the context types identity, environment, location and time. In [Devices Descriptions
for Context-Based Content Adaptation], [Eisinger] [22] most important aspects of context are where you are, who you are with, and what resources are nearby. In [Towards a Better Understanding of Context and Context-Awareness], [Abowd] [7] context type divided into two types: primary context categories and secondary context categories. The primary context types include identity (unique ID), location (position, latitude/longitude), time (duration of events, current time, temporal ordering) and activity (what is happening in the situation). The secondary context categories include all details of the entity and they are also indexed by primary context.

2.4 Context awareness
The term as well as the idea of context awareness have been included in numerous research, and several context-aware application developers state, based on the idea of using context in their application [20], that context-awareness is defined by the ability of the user’s computing device to detect, sense, interpret and response to the context information of the surrounding environment and itself. Dey and Abowd [7] defined context awareness in more general terms as A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task. Therefore, context-aware applications are applications providing services to the users based on the current context of the computing device and surrounding environments.
2.4.1 **Features of context-aware applications**

The main advantage of grouping context-aware applications based on features is to identify the types of applications provided by researchers so far and to provide awareness for context-aware application developers to help them decide what type of features should be included in the design of their application [7][14]. Different papers have tried to categorize the features of context-aware applications, among the first attempts was [Schilit] [11], and [Dey] [7] generally the task is summarized as a matter of obtaining information or executing a command, and whether it is executed manually or automatically.

**Proximate selection application**

Applications obtaining relevant information manually from available context information primarily based on user location are called proximate selection applications [11].
Automatic contextual reconfiguration applications
Applications that deliver relevant information to the user and the reconfiguration of the system, such as adding, removing or changing the connection of components automatically from the available nearby context information, are grouped as automatic contextual reconfiguration [11].

Contextual information and commands application
An application that is used to perform different actions or results manually using the same commands for the user based on the available context. It is grouped into contextual command applications [11] [7].

Context-triggered actions application
An application that is used to perform different actions or results executed automatically based on if-then rules for reacting on context. It is classified as a context-trigger actions application [11] [7].

The second attempt of categorization of features of context-aware applications was made by Pascoe [14]. It is classified as contextual sensing, contextual adaptation, contextual resource discovery and contextual augmentation. According to Pascoe [14] most basic level of context awareness is contextual sensing. It is the capability to recognize contextual information and present it to the user to expand the user’s sensory system. The ability to perform or modify a service automatically based on the current context before providing a uniform service regardless of user context is classified as contextual adaptation [7][14]. Applications that have the ability to locate and utilize resources and services that are appropriate for the user’s context based on the current situation are classified as contextual resource discovery applications [7][14]. Finally, contextual augmentation extends the capability of previous features, sensing, reacting and interacting with the environment, by adding information relating to a particular context [7][14].

The third attempt was made by Dey and Abowed [7]. In general, they tried to describe and summarize previous related works and resulting in three categories: presentation of information and services to a user,
automatic execution of services, and tagging of context information for later retrieval.

2.5 Peer-to-peer computing

Peer-to-peer (P2P) system can be defined as a large self-organizing distributed system without centralized control or hierarchical organization. Each peer in the network has the same capability and responsibility, and all have similar types of software that support sharing resources and services (such as information, network bandwidth, processing cycles, cache storage, processing power or disk storage for files) by direct exchange between systems. One of the key features that differ from the client/server model of distributed computing is that client/server architecture is inherently hierarchical, with resources centralized on a limited number of servers, whereas the P2P model is fully decentralized without any hierarchy. In addition to the decentralization features there are features like self-organization, dynamism, load balancing and fault tolerance; these features provide several advantages such as no central point of failure, robustness, redundancy, shared responsibility, availability and fault-tolerance. This makes it important in areas such as file sharing, distributed social networks, multiplayer online games, and media streaming.

Currently, peer-to-peer (P2P) systems depend on the overlay network protocols for storing, to efficiently search for/locate data items, content distribution, large-scale data sharing, retrieving and routing data or objects. The overlay network protocols can in general be classified as structured and unstructured P2P overlay networks. A structured P2P overlay network means content is placed in a specified location, not at random peers based on the predefined network topology for efficient object discovery. Unstructured overlay networks organize peers in a random topology; new peers joining the network without prior knowledge of the network topology and the network use the flooding mechanism to send queries across the overlay and the receiver peers return a list of all content matching the query to the sender peer. Napster
[28], Gnutella 0.4 [29], Gnutella 0.6 [29], KaZaA [30], JXTA [31] are some examples of P2P overlay networks. Structured P2P overlay networks use distributed hash table (DHT) [32] based protocol to lookup, insert, and delete objects with unique keys in distributed systems. Each peer maintains a routing table consisting of its neighboring peer’s node identifier (node IDs) and IP addresses. Based on the information in the routing table lookup, insertion, deletion and routing are performed across the overlay paths. There are different DHT-based systems: chord [33], CAN [34], pastry [35], Kademilia [36], Kelpies [37] and PGrid [38]. Each system has different organization schemes to maintain routing tables, look up mechanisms and routing strategies.

2.6 Group communication management mechanisms

Nowadays there are many group communication applications on the Internet, in addition to a wide range of social-network group services, such as multiplayer network games, video on-demand, groupware, distributed file systems, video conferencing, replication of services for fault tolerance, and interactive distance learning. Although such services often run on the traditional client/server architecture using the support of multicast, there is quite a number of applications emerging based on peer-to-peer overlays. In applications which provide a group services relying on a peer-to-peer architecture the capability of group formation and group management is considered a very crucial. In social networks people tend to team up with other people with shared interests. Facebook groups, Yahoo groups and groups on Google+ are examples of groups that are quite valued by their users. Groups in social networks could be created based on different criteria, for example humanity, proximity or contexts.

The notion of group in the context of P2P group is defined as any logically connected subset of peers that belong to the same P2P overlay network [40], and “a group communication where the participants can be invited based on context and social networks in both physical and virtual settings” [39]. However, there is a number of research questions that have been raised related to dynamic groups, such as how dynamic
groups can be formed and maintained, how context data can be stored and distributed to support the functionality of dynamic groups, and how the mechanism of group management in peer-to-peer manner. In this work a group refers to a collection of independent entities that can interact, communicate, and collaborate with each other based on their context similarities. Moreover, all members of a group have common attributes, tasks, interests, activities and goals. Each group is organized based on common context criteria, such as interest, location, and different kinds of sensor reading info that can be shared in the group. These context-based groups are dynamic due to the mobility of the nodes in the group. Therefore, group communication management in the emerging peer-to-peer overlays provides a basis for proper handling of such dynamic virtual societies.

Group management is a set of procedures used to create, configure, index, discover, update, and cease groups as well as join, leave and update member nodes in the group. In addition members are capable of interacting with each other and share resource [40]. These procedures could be either user initiated or system initiated. There are several design alternatives available for group communication management in peer-to-peer systems. However, locality aware peer management and formation of groups based on similarity of context can improve the look up efficiency of the peer-to-peer system [9].

A context-based group management mechanism can improve the efficiency of look up by reducing overhead and facilitate dynamic group communication [9].
2.7 The SensibleThings IoT Platform

The SensibleThings Platform [17] is an open source and a fully distributed Internet-of-Things platform that enables sensor based Internet-of-Things applications to share context information and globally access sensors and actuators [3].

![Figure 6: Overview of the function of the MediaSense platform](image)

The above figure shows an overview of how the MediaSense platform functions in the Internet-of-Things application; the MediaSense platform is encapsulated in the architecture and all entities that are connected to the platform have their own MediaSense instance. For example, in the above figure there are three entities, namely sensors, actuators and IoT applications, connected to the platform. Each of them can have their own MediaSense instance to communicate with each other, and each MediaSense instance in the platform is connected to the others via the Internet [3]. The sensibleThings Platform is an implementation and realization of the MediaSense architecture [3].

2.7.1 The SensibleThings Platform architecture

The SensibleThings platform architecture satisfies the fundamental requirements needed by Internet of Things applications to provide adequate QOS and QOE. These requirements include scalability, fully distributed architecture, real-time access, current, bidirectional, seamless, lightweight, stable and extensible [3]. The architecture of the SensibleThings platform is organized into five layers and describes
different layers and components of the platform from top to bottom, including interface layer, add-in layer, dissemination layer, networking layer and sensor and actuator layer.

![Figure 7. The SensibleThings platform’s architecture [18]](image)

**Interface layer:** This layer contains a single component called the SensibleThings application interface, which provides a generic application programming interface (API) for IoT application developers to create their own applications on top of the platform.

**Add-in layer:** This contains two components: extensions and optimization, which enables the addition of extra functionality and optimization algorithms to the platform based on the specific requirements of the applications. In addition, it manages pluggable add-ins dynamically loaded and unloaded in run time when needed.

**Dissemination layer:** This layer enables the distribution of information between all joined entities within the system and platform. This layer implements the primitive operations of the Distributed context exchange protocol (DCXP) to enable the exchange of context information in real time; some of the operations of DXCP include resolving universal context identifier (UCI) and successive direct transfer of context information. This layer contains three components: the dissemination core component, which provides the primitive function of the DCXP, the lookup service component which resolves and stores UCIs within the
system, and the communication component which provides the transport layer communication.

**Networking layer:** This layer is mainly made up of two inner components, an IP networking and physical network medium, which provide the ability to connect different entities to the platform based on Internet Protocol (IP) infrastructure such as wireless or wired, and mobile networks.

**Sensor and actuator layer:** This layer provides different ways for different types of sensors and actuators to connect with the platform. Sensors and actuators can be directly connected if they are accessible through an application code, or they can be connected via the sensor and actuator abstraction component. This sensor and actuator abstraction component provides the ability to connect directly to the sensor and actuator networks, or indirectly via gateways. This layer is organized into five components: physical sensors and actuators, sensor and actuator networks, sensor and actuator gateways, sensor and actuator abstraction, and direct sensor and actuator access.

### 2.8 Distributed context exchange protocol (DCXP)

The distributed context exchange protocol (DCXP) is an application layer protocol that facilitates the reliable exchange and sharing of context information between nodes that have joined a peer-to-peer network [58]. DXCP uses the XML format to exchange data between objects, as well as meet the requirements of the real-time contextual information sharing mode [58].
Context User Agent: is a single endpoint formed by combining source and sinks and allows nodes to join a context network by registering with context storage and enable the exchange of context between sources and sinks [58].

Context storage: uses distributed hash table (DHT) to map between UCIs and source addresses [58].

A - a database Agent  
C - a database client  
D - data miner  
LCUA - Limited CUA  
RCUA - Remote CUA

2.8.1 DXCP naming scheme

The DXCP naming scheme uses Universal Context Identifiers (UCIs), which is similar to the SIP naming scheme format known as URI [58]. Fig.9 provides the syntax and interpretation of UCI.
2.8.2 DCXP messages

DXCP is a SIMPLE-inspired protocol that enables exchange of context information between source and sinks with the five primitive operations REGISTER_UCI, RESOLVE_UCI, GET, SUBSCRIBE, and NOTIFY. These primitive operations of the protocol have been implemented on the dissemination layer of the SensibleThings platform architecture in order to exchange context information in real-time [58].

REGISTER_UCI

A context user agent uses REGISTER to register the UCI of context information with the data storage.

RESOLVE_UCI

In order to find where context information is located, a context user agent must send RESOLVE to the context storage.

GET

Once the context user agent receives the resolved location from the context storage, it GETs the context information from the resolved location.

Figure 9. Universal Context ID’s (UCI) syntax and interpretation
SUBSCRIBE

SUBSCRIBE enables the context user agent to start a subscription to a specified context information, only receiving new information when the context information is updated.

NOTIFY

The source context user agent provides notifications about the latest information to the subscribing context user agents every time there is an update or if asked for an immediate update using GET.

Figure 10. DXCP signaling examples of fetching a context value in the DXCP [58]

2.9 Related work

Nowadays user profile description, modelling, and managing have been the focus of research in industry as well as academia. This section presents related works that focuses on user profile description, modelling and management issues.
Da-morikawa et al. [23] discuss and present with the paper title "A proposal of user profile management framework for context-aware services". They discuss and present a proposed user profile management framework. The framework describes the mechanism of how user related information is combined, updated and disseminated to others to improve the relevance of context-aware services with appropriate privacy control from various sources. They propose a PA (Profile Aggregator) as a node to collect various profiles from various sources and to manage them in the form of a personalized profile. These profile aggregators (PA) also have a role in disseminating a subset of a personalized profile as well as in the query interface, trigger interface and access control mechanism.

To represent the personalized profile data model, add new attributes to the combined user related information, and extend the model dynamically they use the RDF (resource description framework). RDF is a knowledge representation language capable of modeling context information.

A personalized profile is both static and dynamic in nature. The dynamic information should be capable of dynamic data updating. The update method of a personalized profile managed in a profile aggregator has two modules: style reformat module and personalized profile update module. The input XML format profile data is transformed into RDF/XML data format via the style reformat module (XSL) and the update of personalized profile is performed by a personalized profile update module.

In the dissemination of a personalized profile for context-aware services they use query and trigger interfaces. RDQL is similar to the SQL query language for RDF and is used to retrieve target statements or values from the personalized profile. Context based service providers access the personalized profile when receiving the trigger information from the profile aggregator. The user is interested in the privacy control mechanism, the type of data that can be shared, with whom and in what situations.
P. Paula et al. [24] discuss and present with the paper title "Collective knowledge ontology user profiling for twitter". They introduce a scalable and automated technique for user profiling by extracting URLs from publicly available Tweets and using a semantic ontology in which user interests and intensions are characterized. In their proposed approach a user profile is modeled by ontology based on the RDF or OWL (Ontology Web Language) format. The proposed user profiling algorithm discriminates between interesting and uninteresting URLs.

G. Bartolomeo [25] discusses and present with the paper title "User profile management in next generation networks". In this paper the author presents and discusses a proposed solution based on two complementary technologies XDM (XML Document Management) and XDI (XRI Data Interchange) integrated in next generation networks to enable interoperability of user profile data management between telecommunication and Internet services.

2.9.1 WC3 CC/PP standard
CC/PP stands for Composite Capabilities/Preference Profile [19], and is a standard proposed by W3C to express device capabilities and user preferences by providing a vocabulary and a model. CC/PP vocabulary includes identifiers like universal resource identifiers, which are used to refer to specific capabilities and preferences.

CC/PP is an RDF based profiler [19]. It includes collection of information of capabilities of the hardware platform and system software and preferences of the user. RDF is used to create profiles that describe user agent capabilities and preferences.

The evaluation and analysis of this paper [21] identify some of the limitations in the use of the CC/PP model and conclude that CC/PP is not suitable for the representation of a context for pervasive systems, but the solution provided based on XML [22] appears suitable.

2.10 Proposed user profiling
User profiling is the process of building a profile based on a set of data [27]. All information characterizing a user and its environment is called a
user profile. In detail, a user profile is defined as the interpretation of user interests, intensions, characteristics, behaviours and preferences in general. Some of the examples of a user profile are user location, items bought, topic of interests, news sites recommended, and advertisers to serve targeted ads. In recent years, in social networks, user profiling has been widely used to provide useful personalized services by adopting the interests and intentions of users.

S. Ait Chellouche et al. [26] discuss and present with the paper title "Flexible user profile management for context-aware ubiquitous environments ". They introduce a flexible context-sensitive user profile model that gathers static and dynamic data characterizing the user and the user’s operational context to facilitate the adaptation process. They also propose an efficient framework to manage and deliver the user profile. In addition, they present a prototype developed together with IPTV streaming scenarios. The solution proposed aims to share the same user profile and its management between services and several applications by creating a middleware that manages the user profile dynamically and provides applications with all information they could need.

In order to construct and model of a user profile they identify the relevant context information based on the definition of a context as given by Anind K Dey [8] “Context is any information that can be used to characterize the situation of an entity”, and categorize it into five components according to the identified entities, namely general user profile, device profile, network profile, service profile, and context profile.
3 Methodology

This section presents the methodology. The aim of this thesis is been to add capabilities in order to set up context information and user profile management on the top of the SensibleThings platform. These capabilities are intended to not only circumvent the inherent issue with misleading formats, but also to manage interacting users’ context profile on the platform. The SensibleThings IoT platform necessitates a well-structured context-information and user profile management to serve the needs of emerging IoT developers and consumers.

\[\text{Build Knowledge base}\] \quad \rightarrow \quad \text{Goal 1}
\[\text{Context information and user profile management}\] \quad \rightarrow \quad \text{Goal 2}
\[\text{Proof-of-concept App (Android)}\] \quad \rightarrow \quad \text{Goal 3}
\[\text{Evaluation}\] \quad \rightarrow \quad \text{Goal 4}

Figure 11: General approach

To reach the goals related work has been studied to build a solid knowledge base in the area of context-awareness, context information, user profile management, and context management. A proof-of-concept application in the Internet-of-Things scenario has also been developed to prove the concepts. Finally, an evaluation of the results has been conducted and illustrated. The flow chart in Figure 11 shows the sequence
of the goals to simplify and to formulate the methods of each specific goal.

The diagram in Figure 11 illustrates the approach and structure of the tasks of the thesis. Accordingly, the steps following the creation of a knowledge base based on related work have been to develop a context information and user profile management solution extension and to develop a proof-of-concept application that proves the functionality of the developed solution.

3.1 Building a knowledge base

Investigation and thorough study was carried out by collecting reliable material from online resources (for example: IEEE, Springer, the Mid Sweden University digital libraries, eBooks and Google Scholar). In the process of investigation the main target was to obtain knowledge about existing trends in user profile and context information management. Existing mechanisms used to manage user profiles and context information has been located in the literature.

3.2 Context information and user profile management

Design and implementation of context information and user-profile management is the aim of this thesis. The solution was designed using the SensibleThings Platform as core platform and the proposed profile management solution as an extension to the core. The extension has been made to seamlessly integrate with the dissemination layer of the platform and fill the gap. The implementation mentioned later in this thesis refers to the additions in the dissemination layer through the add-in layer of the SensibleThings Platform core.

3.3 Proof-of-concept application

An application for a mobile platform was developed to test the solution proposed. The application developed aimed to be used as proof of concept that was laid down in the context information and user profile management solution. Therefore, the app was expected to show user profile management and context information management in a practical
scenario. The proof of concept application provides a similar service as a distributed chat application with context-aware features.

The application gathers data from sensors and disseminates it to the nodes. The nodes could see the contextual information in addition to the user entered text message. Furthermore, the POC aims to resolve real-world problems by an added sensing capability and for this purpose it uses the accelerometer and GPS information gathered from different nodes. The following real-world problem is considered.

In Ethiopia, a large number of car accidents takes thousands of lives and costs millions in property damage each year. According to the 2014 World Health Organization (WHO) report, car accidents in Ethiopia account for death of 37.28 people out of 100,000. A total of 2.77% of deaths in the country are due to car accidents.

One of the causes for this large number of traffic accidents is speeding. Other causes include driving under the influence. In this POC, we use an accelerometer, GPS and the luminosity of the user’s context to disseminate information about speed, location and luminosity of the traveller to a server, giving an authority access to the information, making it possible for employees at the authority to make observations.

The application I plan to implement has an interface that links to the six profiles presented. All users of this app can send a message plus context information from the device. Furthermore, the app continuously sends information to the simulated server. The simulated server is presumed to be a place where traffic police could observe the location of the vehicle. All information of the connected user is static except for data related to the context profile. In the application the following sensors have been used: Accelerometer, Location Sensor (GPS), Ambient Temperature Sensor and Light Sensor.
3.4 **Theoretical evaluation and performance test**

The usability evaluation method does not have a standard to follow. However, there are two common evaluation approaches. The first approach is known as formative evaluation where design assessments are carried out in each stage of development. The second approach is known as summative evaluation and is done once the whole implementation is complete. The type of test performed in this project is categorized as summative evaluation. The evaluation is completely based on quantitative data analysis from end user interviews.
4 Design and implementation

The SensibleThings platform provides end-to-end solutions for the Internet-of-Things paradigm. Moreover, IoT developers are free to add their extensions on the add-in layer. This chapter presents the implementation of the proposed extension and the accompanying proof-of-concept application. The extension contains two parts, i.e. user-profile management and context-profile management. Furthermore, each extension contains multiple sections, i.e. User-Profile, Group-Profile as well as Device Profile, Context Profile and Network Profile respectively (see Figure 12).

Figure 12. Overview
4.1 The proposed extension

The add-in layer provides a channel for my extension to port-in to the platform and fill the gap mentioned in Chapter one. The proposed extension contains two units: The User-Profile Management Unit and the Context-Profile Management Unit. In general, the proposed extension profiles the data as follows:

Figure 13. UCI modifications

The dissemination layer consists of three main parts: the dissemination core, lookup service and communication (see Figure 4). In its current development, Kelips has been made to provide the platform with the lookingup. The current system relies on a protocol known as DCXP to exchange context information between connected entities. The dissemination core exposes the primitive functions provided by DCXP. Therefore, the specific section of interest is the dissemination core and lookup service.

4.1.1 User-profile management

The user-profile management part of the proposed extension manages the details of the information of the user. This section of the module
contains two main categories of profiling: The User-Profile and Group-Profile.

**User-Profile**

Here, primitive data of the user is profiled. In the profiling the following user data are included: name, address, telephone, age, gender, profile image and profession.

In the *SensibleThings* the base UCI have the following format `uci: //user [:password]@domain[/path[?options]]`. This could be for example `Sileshi.Chukala@miun.se` or `test_user@example.se`.

In the proposed user profiling primitive data are appended as follows:

```plaintext
uci://user[:password]@domian/u_profile/name/{name}
uci://user[:password]@domian/u_profile/address/{address}
uci://user[:password]@domian/u_profile/telephone/{telephone}
uci://user[:password]@domian/u_profile/age/{age}
uci://user[:password]@domian/u_profile/gender/{gender}
uci://user[:password]@domian/u_profile/image/{image_path}
uci://user[:password]@domian/u_profile/profession/{profession}
```

**Group-Profile**

Here, primitive group data are profiled. The group data refers to all groups on the platform that are related to the user. This includes groups the user created, groups the user is a member of, groups similar to the current group profile and user interests. Here, the group profiling has been appended as follows:

```plaintext
uci://user[:password]@domain/g_profile/name/{group_name}
uci://user[:password]@domain/g_profile/interest/{interest}
```
Friends

Here, data of the friends of the users are profiled. In this profile we store friends’ UCI.

uci://user[:password]@domian/group_profile/friends/{friend_uci}

4.1.2 Context-profile management

The term context is defined as in "Towards a better understanding of context and context-awareness" [7]. All information from the embedded sensors and sensory networks are considered context information in this case. The typical example of this is the user location information. The level of context-awareness of a particular application depends of the number of context information resources. Therefore, a special focus of context-profile management has been to enable future additions of context sources without changing the basic structure of the extension.

This section of the module contains three main categories of profiling: Context-profile, device-profile and network-Profile.

Context-Profile

Context-profiling is concerned with sources of context information related with the user. Technically this is a profiling of sensors information. Here, the context-profiling follows a format similar to the user profile case as follows.

uci://user[:password]@domian/c_profile/temperature/{temperature}

uci://user[:password]@domian/c_profile/gps/{gps}

uci://user[:password]@domian/c_profile/accelerometer/{accelerometer}

uci://user[:password]@domian/c_profile/unknown/{unknown}
Device-Profile

Device-profiling is concerned with the device the user uses to interact with the IoT platform. Users could use computers, smartphones, tablets, iPads, smartglasses, smartwatches or some a gadget that has yet to be developed. In this case, the extension profiles the devices as follows:

uci://user[:password]@domian/d_profile/name/[name]
uci://user[:password]@domian/d_profile/category/[category]
uci://user[:password]@domian/d_profile/model/[model]
uci://user[:password]@domian/d_profile/other/[other]

Network-Profile

Network-profiling is concerned with the user communication aspect. The name of the user’s service provider, the medium, and so on. In this case, the extension profiles it as follows:

uci://user[:password]@domian/n_profile/name/[name]
uci://user[:password]@domian/n_profile/medium/[medium]
uci://user[:password]@domian/n_profile/other/[other]

4.2 proof-of-concept application

The application aimed to prove the practicability of the concepts is developed for the Android platform. The main reason for choosing the Android platform is the fact that the core of the SensibleThings platform is available in Java and the fact that Android development provides the cheapest of all competitive technologies. The implementation in this section strives to use every part of the modules that have been proposed and implemented.

The application contains three parts as showed in Table 1, 2 and 3 each of the menus represent the main page which provides an environment much like a chat application, including the user profile and context
profile. So the user section contains: User Profile, Group and Friends. The context section contains: Context, Device and Network.

Table 1. User profile contents

<table>
<thead>
<tr>
<th>Attribute</th>
<th>User</th>
<th>Group</th>
<th>Friends</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute-1</td>
<td>Name</td>
<td>Name</td>
<td>List of friends</td>
</tr>
<tr>
<td>Attribute-2</td>
<td>Address</td>
<td>Interests</td>
<td></td>
</tr>
<tr>
<td>Attribute-3</td>
<td>Telephone</td>
<td>Number of members</td>
<td></td>
</tr>
<tr>
<td>Attribute-4</td>
<td>Age</td>
<td>Groups created by the user</td>
<td></td>
</tr>
<tr>
<td>Attribute-5</td>
<td>Gender</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute-6</td>
<td>Profession</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute-7</td>
<td>Profile picture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Context profile contents

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Context</th>
<th>Device</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute-1</td>
<td>Temperature</td>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Attribute-2</td>
<td>GPS</td>
<td>Category</td>
<td>Medium</td>
</tr>
<tr>
<td>Attribute-3</td>
<td>Linear acceleration</td>
<td>Model</td>
<td>Other</td>
</tr>
<tr>
<td>Attribute-4</td>
<td>other</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute-5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute-6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Passenger is only required to turn on mobile data, the app sends information to the server when the vehicle passes x speed limit. Other nodes exchange contextual and user information each other. Authorized Person fetches information on the vehicle moving over limit and trace the Location. Servers are used to live log reports from the applications.

**Figure 14. Proof-of-concept app**

*Smartphones contain embedded GPS sensor that could be used to locate the exact location of the device in real time. In Figure 14 a passenger using the POC app is in the car. If the passenger believes the vehicle is exceeding speed limits and wants to report location of the vehicle to the server owned by the authorities. The passenger only turns on the app enabling authorities to trace the vehicle. Furthermore, the passenger could exchange contextual information with other nodes.*

### 4.3 Design summarization

**Requirement one**

Develop an extension which adds functionality for context information and user profile management for the *SensibleThings Platform*.

**Details**

To create our own add-in for the SensibleThings platform (i.e. to add extension)

The extension will force the user to register a base UCI that contains a list of all the sensors of that person.
The extension will force the user to include a list of his friends to enable relationships between the users.

This will enable the extension to prioritize one's friends in order of context proximity.

The extension will also be able to find new friends by looking at the friends of friends.

**Approach**

**Concept:** Consider each node has two unique UCIs.

The 1<sup>st</sup> UCI $\rightarrow$ base UCI $\rightarrow$ simple UCI, unique and the same UCI as the current.

**Table 3. Base UCI**

```
Silshi_chu2/miun.se
```

The 2<sup>nd</sup> UCI $\rightarrow$ extended UCI $\rightarrow$ contains a long string of characters, incorporating the base UCI and contains all the information (user information and context information, friends).

**Table 4. Sample extended UCI: Character length 1208**

```
Silshi_chu2/miun.se/~
Silshi_chu2/miun.se/u_profile/name/Sileshi Chu.:  
Silshi_chu2/miun.se/u_profile/address/Silshi_chu2/miun.se:  
Silshi_chu2/miun.se/u_profile/telephone/+46111111:  
Silshi_chu2/miun.se/u_profile/age/27:  
Silshi_chu2/miun.se/u_profile/gender/Male:  
Silshi_chu2/miun.se/u_profile/profession/Student:  
Silshi_chu2/miun.se/u_profile/profile_image/null~  
silshi_chu2/miun.se/g_profile/name/Test Group:  
Silshi_chu2/miun.se/g_profile/interest/Test:  
Silshi_chu2/miun.se/g_profile/model/null~  
Silshi_chu2/miun.se/c_profile/tempature/20C:  
Silshi_chu2/miun.se/c_profile/gps/39.8929,34.2390:
```
Fulfilment analysis

In the previous section in this chapter, I presented the design of the extension for SensibleThings Platform. The extension was designed so that user profiling and context profiling could be managed on the IoT platform in a manner that could be useful for other IoT developers working on the SensibleThings Platform. The UCI has been modified so that all the user profile, group profile, friend profile, context profile, device profile and network profile could be fetched (see Section 4.1).

The following UCI will provide a link for all the friends:

\[ uci://user[:password]@domain/f\_profile/friends/{friend\_uci} \]

If there is a mechanism to access all friends, then it is up to the upper layer (or application layer) to fetch all information about friends of friends.

The extension provides the context profile in the following format:

\[ uci://user[:password]@domain/c\_profile/{context}/{context} \]

Therefore, if all connected users are in this format of context profiling, then the application layer could work on the values of contexts to carry out prioritization.
The extension has been designed so that connected users can provide information of their context profile, user profile, device profile and network profile. Therefore, the application could search friends based on different keys from these categories.

**Requirement Two**
Develop IoT proof-of-concept application that consumes these extensions and runs on top of the *SensibleThings* Platform.

All codes must be implemented from scratch.

**Approach**
Implement test application on the Android platform.

**Fulfilment Analysis**

The android application was designed to be as minimal as possible yet including all parts of the proposed profile management extension. Furthermore, the proposed solution and the proof-of-concept were implemented from scratch according to the requirement so all codes in this project have originated from the author of this paper.

**Requirement Three**
Evaluate the project outcome and illustrate the overall gain of the solution proposed.

Evaluate the results and propose future work

**Fulfilment analysis**
This goal is presented in the results chapter.
5 Results

This chapter presents the implementation results and the output of the usability survey. The main aim of this project has been to propose an extension which could fulfil the requirements presented in section 1.5. Furthermore, in order to make sure the extension can be used in a practical scenario, a test application has been implemented. To make sure this could be acceptable in the community, a survey has been carried out. The aim of the survey was to obtain information about the usability of the applications based on the extension developed. The result of each of these sections was presented in sections 5.1, 5.2, and 5.3 respectively.

5.1 The proposed extensions

The proposed extension contains two sections as shown in the library bellow. Context Profiling and User Profiling:

<table>
<thead>
<tr>
<th>Package: ContextProfiling</th>
<th>Package: UserProfiling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class: ContextProfiling</td>
<td>Task: Context Data</td>
</tr>
<tr>
<td></td>
<td>{Sensors</td>
</tr>
<tr>
<td>Class: UserProfiling</td>
<td>Task:Handles information about the user</td>
</tr>
<tr>
<td>Class: Devicerofiling</td>
<td>Task:End User Device Information</td>
</tr>
<tr>
<td>Class: NetworkProfiling</td>
<td>Task:Network in use information</td>
</tr>
<tr>
<td>Class: ProfilingExtension</td>
<td>Task:Loads all the platform information onto the platform through the add-in layer</td>
</tr>
<tr>
<td>Class: GroupProfiling</td>
<td>Task:Group information handling.</td>
</tr>
</tbody>
</table>
The design chapter explains that the extended UCI carries all the information, and any node which receives the extended UCI could extract the information incorporated in the long array of characters. However, all
nodes that are a part of the platform also store information such as updated list of friends on shared preferences (see classSharePreference.java (se.sensiblethings.addinlayer.extensions.pm.contextprofiling) in figure 9).

One of the basic requirements in the functionalities of the extension was user friends and user’s friends’ friends browsing capability. The SearchFriends.java takes care of this core functionality. This was based on the advantages of the extended UCI. The extended UCI contains a list of all friends base UCI’s. Each base UCI is mapped to extended UCI and each extended UCI of the user friends contains the friend information; this recurs indefinitely as long as each user has friends.

5.2 The proof-of-concept application

The concept on which the implementation is based has been tested using an application running on top of the Android platform and including the extension explained in section 5.1. Moreover, due to some bugs encountered when using RUDP plus KELIPS for the lookup for nodes not behind NAT, I switched to another simpler application running on Java alone which extracted information and communicated message without difficulty. Finally, since the Android app works perfectly on PROXY_RUDP and KELIPS, I switched to the app in the final testing. This issue is explained in section 6.1.
A Move Towards Practicability -
Silesi Chukala

5 Results
2016-11-27

Figure 17. POC application
5.3 Usability test and evaluations

The type of evaluation carried out in this section is categorized under summative evaluation. The evaluation is based on the quantitative survey.

5.3.1 Survey parameters

The subjects involved in this empirical data analysis are people from various parts of the world who are familiar with popular smartphone applications like Facebook, Viber, Google+, WhatsApp, Tango, and Skype like. Some of the subjects involved in the empirical data analysis study have the knowledge to be able to provide an expert review of the application. For the survey, one hundred volunteers were chosen to provide feedback on the test application. Each volunteer was provided with a list of questions about the app (Appendix A) to answer. Finally,
5 Results

Move Towards Practicability -
Sileshi Chukala

2016-11-27

each volunteer was given a chance to rate the proof-of-concept application.

5.3.2 Data collection approach

The survey aimed for one hundred participants from countries around the world that could install test application and fill out the forms they were provided. The usability test carried out in here is considered a Remote Usability Testing, as opposed to a Hallway Testing, where the test is carried out in a limited geography.

The link for the test application and the accompanying form was given to each participant. All participant data are examined thoroughly and the graph in Figure 14, representing the results.

![Figure 19. Results](image-url)

In the survey, the subjects who were all familiar with social media provided their answers to questions on Grouping and User Profiling, Friend Extension, Context Profiling, and User Profiling. According to the answers, all of the subjects believe that Grouping is important while 95 of them answered they only want to see their immediate friends, there is no need for an extended friend browsing capability. Furthermore, 67 of the subjects believe it is important to have context profiling on the application, while 17 mentioned that since there are other sources such
as sensors they do not find it important because it will show private information such as location. Furthermore, more than 10 participants failed to understand the issue.

5.3.3 Technical comparison

In the proposed extension the size of the UCI increases in size as the information from a particular node increases, as well as when the number of friends grows. This type of approach could have devastating impact on the performance of the platform, especially as churn size increases at the same time as other context/user information. In this section, a comparison has been made of the basic approach.

Table 5. Comparison of the platform with add-on and without add-on

<table>
<thead>
<tr>
<th></th>
<th>Data Size for Base UCI</th>
<th>Data Size for Extended UCI</th>
<th>Resolve Time</th>
<th>Friends of Friends Extensibility</th>
<th>Capability to Form Group</th>
<th>Capability to Add Even More Contextual Information</th>
<th>Users Satisfaction [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing</td>
<td>Minimal</td>
<td>Not applicable</td>
<td>&lt; 1 second</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>[Mentioned the extension doesn’t matter]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existing System with Proposed Extension</td>
<td>Minimal</td>
<td>Increases as the number of peers and contextual information increases</td>
<td>&lt; 1 second</td>
<td>Provides</td>
<td>Provides</td>
<td>Provides</td>
<td>100[Mentioned the extension matters]</td>
</tr>
</tbody>
</table>

In addition, to gain knowledge about the impact of the extended UCI on the performance of the SensibleThings platform, an observation has been made where the size of UCI was increased to 1024KB; the test revealed no considerable delay or any negative impact on the platform. Note: UCI of 1024KB was been taken as a maximum in this investigation due to the fact that the base UCI only uses a few bytes and this UCI size is considered quite large.
6 Conclusions

The project goals listed in the first chapter were approached as stated in the methodology chapter. In general, according to the results presented in Chapter 5, the project goals were successfully achieved. In light of the projects goals presented in Chapter one, the following was achieved:

**Goal One:** *Build a solid knowledge base needed to successfully carry out the aim of this thesis.*

A literature review was carried out in Chapter 2. In the chapter, related work, technology, emerging trends, and current problems were presented. This prepared the author to carry out the following parts of the project.

**Goal Two:** *Design and develop functionality for context information and user profile management for the IoT platform.*

Design and development of the extension was successfully carried out according to the minimum requirements mentioned in the project proposal. The corresponding design was explained in Chapter 4 and 5.

**Goal Three:** *Develop an IoT proof-of-concept application that uses these extensions and runs on top of SensibleThings Platform.*

The application including the extension developed on the add-in layer of the IoT was implemented for Android devices. The proof-of-concept application uses the extension to show the advantages and the contribution to the existing implementation.

**Goal Four:** *Evaluate the project outcome and illustrate the overall advantage of the solution proposed.*

The results, quantitative survey results, and the comparison and performance examination prove that the proposed extension could push the platform to the next level by some degree.
6.1 Challenges and coding errors

In the implementation stage I came across a situation where two nodes communicating in the Local Network remained busy indefinitely. This was the major challenge during the test of the Android emulators. The same code works correctly for Java applications and the testing with the emulator works correctly for nodes connecting through the PROXY_RUDP relay. So in this case I developed a code for a simpler localclient and localbootstrap nodes, then the extended UCI and base UCI concept was tested before moving on to test the same concept on the PROXY_RUDP using the emulator.

![Figure 20. POC challenge and coding error](image)

6.2 Discussion

The results from the tests carried out with the proof-of-concept application suggest that the extension could be used for the intended purposes. Furthermore, the usability test carried out to find out about the idea of
context and user profile management as well as friend browsing capability suggest that the extension is not at risk of rejection in future deployments.

6.3 **Contribution and impact**

The contribution of this thesis work could be considered double; i.e. from the point of view of end users and from the point of view of Internet-of-Things developers. Consumers could employ the user and context management extension proposed without any further modification of the technique itself. The extension developed in this project provides practical, feasible and distributed profile management and browsing capability. Moreover, developers could modify this concept further and use the proposed extension as a stepping stone for a better solution.

6.4 **Ethical considerations**

The proposed approach for context and user profile management in this thesis could be consumed by end users for practical applications. The proposed approach could also serve as a stepping stone research in related projects. The results from the test carried out with the proof-of-concept application plus a usability test proved that the technique could be employed directly in a practical scenario.

In terms of regulations and rules such as copyright of material used in this thesis and similar related issues related to plagiarism, all the material and references mentioned in the report was properly addressed in compliance with the copyrights rules and standard reference style of Mid Sweden University.

6.5 **Future work**

Possible future works related to the concepts implemented in this thesis are the following:

To propose a general approach for user and context profiling that could work for similar IoT platforms.
To examine the drawbacks of the approach for context and user profile management.

To make the extension as convenient as possible for future consumers. Here, additional polishing and debugging tasks are important.
References


[8] K. Dey, ”Understanding and Using Context”, College of Computing and GVU Center, Georgia Institute of Technology.


[29] S. Ertel, “Unstructured P2P networks by example: Gnutella 0.4, Gnutella 0.6”. Current topics in dependable distributed systems, Systems Engineering Group Dresden University of Technology.


A Move Towards Practicability -

References

2016-11-27


Appendix A:

Questions I –

1. For how many years have you been using the Internet, Social sites and smartphones?

2. Can you write three of your noble social apps you use each day? Please write in chronological order. Are you familiar with Facebook, Whatsapp, and?

3. If yes, do you find that there are similarities between any of these applications and the test application you are running? Do you see any unique advantages with this app compared to other related applications you have used before?

4. How many different information categories are you able to fetch? Did you face any difficulties accessing each of the activities? Did you come across anything confusing and inconvenient? Did you came across any delay when fetching information?

5. Please list any the difficulties you came across in each these activities

   Difficulties with User Profile Access:

   Difficulties with Context Profile Access:

   Difficulties with Network Profile Access:

   Difficulties with Profile Access:

   Difficulties with Group Profile Access:

   Difficulties with Friend profile Access:

6. How do you rate this test application?

   - 1
   - 2
   - 3
   - 4
   - 5