Title of thesis

Subtitle of the thesis, if any

Leon Hennings
Kamyar Sajjadi

Department of Computer and Systems Sciences

Degree project 15 HE credits

Degree subject: Computer and Systems Sciences

Degree project at the Bachelor level

Spring/Autumn 2013

Advisor: Rahim Rahmani

Assistant advisor: Jaime Walters

Reviewer: Name of Reviewer

Swedish title: Uppsatstitel

Stockholm University

Title of thesis

Subtitle of the thesis, if any

Leon Hennings
Kamyar Sajjadi

Abstract

Qui terempri isuam liam incum aris forteat strudentre quid inculoc iviliaet; nihilius. Catimor demquam tam, senterf rmactum ompec orum, ne culvitimis se constusquam pratastiam porum rem, que poptert batia vidiem, sendum patus? Paturo vis esum inessus, furnimi iliciem ent vidit rebaturei potiu quam ductame teris, conductam me consum retis consuniu ma, speriam rem firisum publi, quam factod norehem et grat. Maesse, se orteliciemus cem. Fuidena ilicien ervid C. M. C. me quid rem moente, conimus nit practanti, tabena, Ti. Habunterfer que avoltum quam hinaterniam abem nonside labistiae, intelic vidi ium erce enatici ecturox nulviribem hostio nesimoeritia rei inatis.

Keywords

Keywords for the thesis, common in the domain.

Contents

1. Introduction

1.1 Background

Around us today is a large assortment of devices with sensors which generate data. This data is mostly used locally or together with a other devices in a small system. Smartphones and mobile broadband allow people to have access to internet wherever they are. This access to the internet in order to share information is not limited to people. Sensors from different devices can be connected to each other and communicate with Internet Protocol [5]. This is resulting in an Internet of Connected things, where each thing, whether human or machine can connect and communicate, sharing and digesting information, executing tasks and collaborating to realise massive Immersive Environments [25]. This builds a new network called the Internet of Things [25].

The Internet of Things, aims to seamlessly fuse people places and things across current communications platforms, realising immersive situations that are enabled through the collection of information from embedded sensors and respond by acting upon corresponding embedded actuators.

Sensors embedded within our environment range from simple sensors such as temperature, humidity, light intensity and occupation sensors, to location and GPS sensors embedded in mobile devices, telephones and automobiles. Applying approaches such as sensor fusion allows us to emulate even higher level sensor information exposing information that is otherwise not directly observable. All these sensors collects data, and with this data developers can build applications that act based on the given data from the sensors. These could range from an application that can automatically regulate the heating in your home to an application that can tell you which way you should take to work according to the traffic on streets.

In order to enable these behaviors and realise the impending immersive paradigm, sensors of a device need to collect and share context information.

Earlier research in the area realised the use of middleware systems for this large scale information provisioning. These included both centralized and decentralized approaches.

Centralized approaches such as the IP Multimedia subsystem[ref] and XXXXX, exist as web service portals on the internet, providing a point of connection for entities to provision context information on an Internet of Things. However, these approaches assume the complete availability and reliability of such systems which are susceptible to DNS errors, denial of service attacks and dynamic IP configuration issues. They are also prone to creating bottlenecks which affect real time information sharing which means that we are not able to guarantee information accuracy. This is important for an Internet of Things where real-time is key to creating Immersive participation environments. Fur-

thermore, the scalability is limited when using a centralized approach [17]. The problem with scalability and DNS availability makes centralised solutions suboptimal.

In order to fix this it was suggested we use distributed approaches, these do not rely on DNS and are more scalable and not prone to DOS attacks. Such approaches include MediaSense [17] and UBIWARE [21] and these have both the "server" and "client" on one machine. Within the MediaSense realization, applications are tied to the platform and are started and terminated with the platform. Every application has its own MediaSense instance and communicates with other devices through this instance.

Another thing that needs to be considered when developing middleware for the concept internet of things is that devices in our everyday life has limited resources. The current way applications run on MediaSense, where every application needs its own instance of the platform, makes it very inefficient.

1.2 Problem

Distributed Internet of Things middlewares are designed for resource capable devices such as desktop computers and servers. One of the main requirements for Internet of Things middleware defined by Theo Kanter et al. [17] is to have a lightweight middleware. Since the sale of smartphones surpassed that of computers and laptops in 2011 [reffy] the Internet of Things will heavily incorporate more ubiquitous devices with lower resource availability that can be mass deployed. The current design of distributed Internet of Things middleware excludes usage of such ubiquitous devices.

1.3 Goal

1.4 Scope

2. Extended Background

Enabling a middleware that supports an Internet of Things requires and understanding of the theories and concepts that support and maintain this connected things paradigm. In this chapter we look at what the Internet of Things is and the surrounding concepts that is used for Internet of Things. We will discuss what context aware applications are and why context information is important for this concept. The architecture of MediaSense will also be described so the problem statement will be more clear.

2.1 Immersive participation

Immersive participation is focused on participation on the Internet via ubiquitous computing and context awareness. They enable people, places and things to connect to each other to create Immersive Participation Environments. Immersive Participation Environments provides users with context-awareness everywhere which makes the users participate as if they are in a virtual world with places, things and people in it. Common examples of Immersive Participation Environments today include Google Ingress [20] where users join teams and compete with other teams in a virtual world where they need to take over real world artefacts, TURF [1] where users capture real world places and gain points and RATS Theatre's [26] application called Maryam [8] which is an interactive theatre where audio clips is triggered depending on the users GPS location. Larger scale immersive applications will benefit from scalable distributed information sharing and also remove bottlenecks and dependencies on centralized web portals on the internet. The way humans interact with each other and things around them will change when sensor information can be shared and accessed ubiquitously. Creating immersive environ that blend the natural world with a seamless internet of things require that we are able to understand the situation of the users in real-time this understanding is termed context awareness.

2.2 Context Awareness

Software developers aim to build applications and systems that are easy to use. A way to accomplish this is to not only use data given by the user but also use context information from the users environment. Utilizing context in applications is called context-awareness.

When a person talks to another person they are able to share specific information or context to increase the richness, depth and meaning of the conversation. The situation and the context makes it possible for a human to act according to this information. An example of this is when a person is asking another person for directions to a place. The context in this conversation can be the location the persons are standing at and based on this the person giving the directions can either point or give a description of the sequence

of right and left turns needed to reach the destination. By having this context information it is possible to give directions to the desired location. This kind of ability is therefore hard to transfer to human interaction with computers. Improving the computers ability to access and understand the circumstances developers can build context-aware applications that are easy to use. In order to understand what context-awareness is and where it is used, we first need to get a clear definition of context. In earlier work Schilit and Thimer [23] is defining context as locations, identities of close people and objects. This definition is too specific. It's not only locations that is interesting as context. A better definition of context is:

> "Context is a combination of any information that can be sensed or received by an entity which is useful to catch events and situations. [7]"

In other words context is information from an entity that gives specific information to increase the understanding of an events environment. An entity can be a person, place or a object that is relevant for the interaction. A typical example of a device capable of generating this type of information is a smartphone.

One such smartphone is the a Google Nexus 4 [13] which contains, among other things, the sensors an accelerometer to detect acceleration, a GPS to receive location data, a gyroscope to detect rotation, a barometer to detect air pressure and a compass for direction and navigation. By applying sensor fusion [ref] other context data can be attained. An example of this would be combining a barometer with GPS to detect altitude[ref]. Using these types of information, we can realise applications that are context centric. Similar to the earlier example where a person was asking another person for directions to a place, the person can now ask for direction from his phone. The phone will collect context-information from the its GPS sensor and based on this location be able to give the directions.

Context-awareness in applications will be more interesting to use when the concept Internet of things is growing. It's important for applications to take part of the context. When context is changing the applications need to handle the changes to fit the users interaction. Schmidt defines context-awareness in this way:

> "Context-awareness is the ability of an entity to usefully adapt to or react based on context. [7]"

Another definition of context-awareness is:

> "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. [3]"

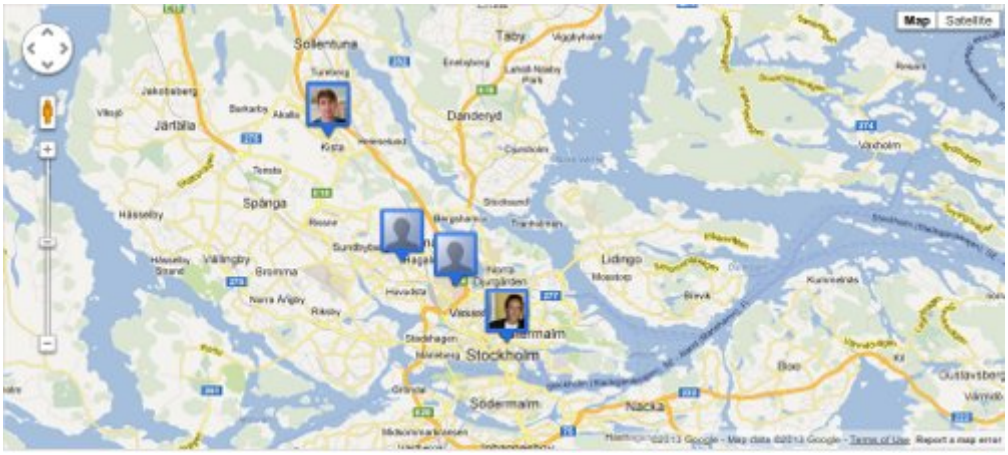

Figure 2.1: A picture of Google Latitude showing contacts shared positions.

Figure 2.2: Android homescreen on a Sony Ericsson Xperia PLAY rotates based on context information from its sensors when the phone is physically rotated.

With these two definitions context-awareness means that we can use the context that is received by an entity and uses this information to interact with the application. An application is a context-aware application if it is able to use context in order to adjust its behavior or the content it is providing. Example of context-awareness in applications is Google Latitude[7]. The application makes it possible for your friends to see your position on a map. Google Latitude uses the locations from a entity (mobile phone or computer) to update your position on the map. Another Context-aware application from google is AdSense/AdWords. This application generates advertisement based on the context of the webpage the user have visited. Also Android is a good example of context-awareness in application. When a user rotates a smartphone or a tablet the operating system changes its orientation to landscape mode.

Context awareness can change classical scenarios into intelligent responsive scenarios by using context informations to behave in a special way. Common applications such as home automation can use information to turn on the light at home. In earlier implementations where applications did not use context information and users turn on lamps by

manually pressing a button. In comparison, if the developer instead uses the context information the application can interact on information from sensors around the user. For example when a user enters his home the light will be turned on when the application detects his location from the users ubiquitous device. The developer can also use information that provides how the weather is outside, if it's sunny outside the application will act to this and it will not turn on the lights when the user enters his home, but if it's cloudy outside the application will turn on the light. Context awareness on a massive scale is gradually enabled by the advances in pervasive and ubiquitous computing.

3. Method

Bibliography

[1] Andrimon AB. Turf - outdoor addiction. http://www.turfgame.com/. Accessed: 2013-05-12.

[2] Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Zoran Despotovic, Manfred Hauswirth, Magdalena Punceva, and Roman Schmidt. P-grid: a self-organizing structured p2p system. *ACM SIGMOD Record*, 32(3):29–33, 2003.

[3] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[4] Ilia Bider, Paul Johannesson, Erik Perjons, and Lena Johansson. Design science in action: Developing a framework for introducing it systems into operational practice. AIS, 2012.

[5] Michael Chui, Markus Löffler, and Roger Roberts. The internet of things. *McKinsey Quarterly*, 2:1–9, 2010.

[6] U.S. Election Assistance Commission. Definitions of words with special meanings. http://www.eac.gov/vvsg/glossary.aspx. Accessed: 2013-05-14.

[7] Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.

[8] Dramaten. Maryam. http://www.dramaten.se/Dramaten/Forestallningar/Maryam/. Accessed: 2013-05-12.

[9] Albrecht Fehske, Gerhard Fettweis, Jens Malmodin, and Gergely Biczok. The global footprint of mobile communications: The ecological and economic perspective. *Communications Magazine, IEEE*, 49(8):55–62, 2011.

[10] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[11] Nahid Golafshani. Understanding reliability and validity in qualitative research. *The qualitative report*, 8(4):597–607, 2003.

[12] Google. Google latitude. www.google.com/latitude. Accessed: 2013-04-29.

[13] Google. Google Nexus 4 tech specs. http://www.google.com/nexus/4/specs/. Accessed: 2013-04-29.

[14] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.

[15] P. Johannesson and E. Perjons. *A Design Science Primer*. CreateSpace Independent Publishing Platform, 2012.

[16] T. Kanter, P. Osterberg, J. Walters, V. Kardeby, S. Forsstrom, and S. Pettersson. The mediasense framework. In *Digital Telecommunications, 2009. ICDT '09. Fourth International Conference on*, pages 144–147, 2009.

[17] Theo Kanter, Stefan Forsström, Victor Kardeby, Jamie Walters, Ulf Jennehag, and Patrik Österberg. Mediasense - an internet of things platform for scalable and decentralized context sharing and control. 2012.

[18] Artem Katasonov, Olena Kaykova, Oleksiy Khriyenko, Sergiy Nikitin, and Vagan Terziyan. Smart semantic middleware for the internet of things. In *Proceedings of the 5-th International Conference on Informatics in Control, Automation and Robotics*, pages 11–15, 2008.

[19] Bruce Jay Nelson. *Remote procedure call*. PhD thesis, Pittsburgh, PA, USA, 1981. AAI8204168.

[20] NianticLabs@Google. Ingress. http://www.ingress.com/. Accessed: 2013-05-12.

[21] Hubert Österle, Joerg Becker, Ulrich Frank, Thomas Hess, Dimitris Karagiannis, Helmut Krcmar, Peter Loos, Peter Mertens, Andreas Oberweis, and Elmar J Sinz. Memorandum on design-oriented information systems research. *European Journal of Information Systems*, 20(1):7–10, 2010.

[22] Matthew D Portas, Jamie A Harley, Christopher A Barnes, and Christopher J Rush. The validity and reliability of 1-hz and 5-hz global positioning systems for linear, multidirectional, and soccer-specific activities. *International Journal of Sports Physiology and Performance*, 5:448–458, 2010.

[23] Bill N Schilit and Marvin M Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, 1994.

[24] Roman Schmidt. The p-grid system-overview. *School of Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne (EPFL) CH-1015 Lausanne, Switzerland, Tech. Rep*, 2007.

[25] Lu Tan and Neng Wang. Future internet: the internet of things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 5, pages V5–376. IEEE, 2010.

[26] Rats Teater. Rats teater. http://ratsteater.se/. Accessed: 2013-05-12.

[27] Jamie Walters, Theo Kanter, and Enrico Savioli. A distributed framework for organizing an internet of things. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, pages 231–247, 2012.

[28] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, July 1999.

[29] Laurie A Williams and Robert R Kessler. All i really need to know about pair programming i learned in kindergarten. *Communications of the ACM*, 43(5):108–114, 2000.

Stockholm University

Department of Computer and Systems Sciences

Forum 100

SE-164 40 Kista

Phone: +46 8 16 20 00

www.dsv.su.se

Stockholm
University