Context-awareness and context modeling

Ubiquitous Computing Seminar FS2014

Student report

Sandro Lombardi
sandrlom@student.ethz.ch

June 10, 2014

ABSTRACT
This student report is intended to give an overview over the topic of context-aware applications and context-modeling. After starting with a motivation, some fundamentals about context and context-awareness will be covered. In a next step, a live cycle of context in context-aware applications is shown, which among others covers context modeling. Finally, some research papers are presented which have tackled the problem of physical activity recognition, an important part of context recognition.

Keywords: Application support, Context, Context modeling, Context-awareness, Situation-awareness, Context modeling.

INTRODUCTION

Context-awareness, as a core topic of ubiquitous computing, promises improvements from various perspectives. As from the Internet of Things perspective, the number of sensors deployed in our daily life and the world will increase at a fast rate. Since context-aware computing has proven to be successful in understand sensor data, it plays an important role in this area. But also the area of Human-Computer-Interaction can profit from context, since naturally, humans communicate by implicitly using context. By improving the access to context, the communication between human and computer can be enriched. Personalization as another perspective uses context to offer relevant, context sensitive information to users with mobile consumer electronic devices.

The topic of context-awareness is closely related to other topics in ubiquitous computing, as context shows up in many different ways:

1. Energy can be considered as context in the research area of smart energy
2. In smart heating, context like occupancy, time and location play an important role
3. Physical activity recognition which is important in context recognition, can for example be achieved through smart glasses.

Context promises enhancements for applications and we call such applications, which make use of context, context-aware.

Why make use of context?

Access to context means that applications are able to better understand their environment, its user and the current situation, such that they can react appropriately and provide situation-specific solutions and support. A mobile tour guide application can for example make use of location and provide relevant information, while a user with a smart-phone can spontaneously make use of the smart-TV to display things on a larger screen. While context makes less sense for fixed-location computers which are acontextual in contrast, the nature of mobile devices is contextual. Context-based retrieval applications gather and store context information and allow later information retrieval based on context information. For example, a user can ask a note-taking application to show the notes associated with a previous meeting for the current meeting. Beside this, some more improvements, context-awareness can offer involve Machine-Machine-Interaction and other mentioned perspectives. Thus, a washing machine might communicate with a heating system to reduce peaks in energy consumption from an Internet of Things perspective, while for example Google shows how the use of context in form of web-search history can improve personalization of applications.

CONTEXT FUNDAMENTALS

The notion of context existed long before its use in ubiquitous computing. The word ”context” comes from the Latin word ”contextus” which means coherence or connection. Many researchers have proposed several definitions for the term context. Through a definition, application developers are able to tell whether some piece of data is context or not, simply by applying the definition. Schilit and Theimer [5] referred to context as location, identities of nearby people and objects, and changes to those objects. Dey et al. [1] claimed that the definitions given by Schilit and Theimer were based on examples and provided a own definition:

”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.” Other definitions have used synonyms, such as for example ”‘situation”’ or ”‘environment”’.

The term ”‘context-aware computing”’ was first introduced by Schilit and Theimer [5] as software, which ”‘adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time”’. In [1], Dey et al. provides an alternative, applicable and widely accepted definition for this term:

”A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the users task.”

Characteristics of context

To see the difficulties in most context-aware applications, observations have been made about the nature of context information in pervasive computing systems.

1. Context must be abstracted to make sense. Most data comes from sensors and what one has at this point, is raw context data, which must be processed further to make sense for the end user. For example one might get raw GPS coordinates out of the the GPS-sensor device, but what really is needed is the geographical location like the name of a city.

2. The sensors of which context may be acquired from can be distributed and heterogeneous. Considering a smart home scenario as an example, the context information about the physical activity of the user may be acquired by the door-opening sensor of the oven and a passive infrared-camera. This also introduces the problem of the next point.

3. Context has many alternative representations. This makes it harder since in the process of context data fusion, one has to consider whether two information pieces represent the same context or not.

4. Context is dynamic, which means that time and place can change the acquired context. There does exist static context, like the identity of a user for example, if the identity is assumed to never change, but most context has a dynamic character.

5. Context information is imperfect and uncertain. The acquired sensor data may contain errors or be unstable. To counter this characteristic, missing values in sensor data have to be filled, outliers must be removed and the data must be validated through use of different sources.

Features of context-aware applications

There have been approaches to identify features of context-aware applications on an attempt to define further the notion and the field of context-aware applications. Abowd et al. [8] compared the two approaches of Schilit et al [6] and of Pascoe [7] and combined them to a now widely accepted categorization:

1. The presentation of information and services to a user. For example, a mobile application can dynamically update a list of closest printers as its user moves through a building.

2. Automatic execution of a service, which means for example that the application automatically prints the document at the closest printer to the user.

3. Tagging of context to information for later retrieval is the third point. For example, an application can record the names, the times and the related printer of the printed documents. Later on, the user will be able to retrieve this information to find his forgotten printouts.

Levels of context-awareness

Related to the features of context-aware application Barkhuus and Dey [9] differentiated levels of context-awareness based on the user interaction. The three levels are personalization, passive and active context-awareness.

1. Personalization covers the level at which the user is allowed to set preferences, likes, and expectation, i.e context information is set manually only. Consider an example, in which the user may set the preferred temperature in a smart home environment. Such an application would be classified into the level of personalization.

2. With passive context-awareness, the application on the other hand monitors the environment and offers appropriate options to the user, i.e it can be considered as semi-manually user input. For example, if a user enters the super market, his mobile phone may alert the user with a list of discounted products.

3. Active context-awareness lets the application continuously and autonomously monitor the situation and in addition act autonomously. As an example, smoke detectors and temperature sensors can detect a fire in a room in a smart home environment and as reaction to that, the system notifies the fire brigade and the owner of the house.

Context categorization

To help application designers find the most likely pieces of context that will be useful in their application, different categorization schemes have been proposed based on different perspectives.

Abowd et al. [8] proposed to identify location, identity, time and activity as primary context types, since in practice, they were more important and these four types of context answer the questions of who, what (i.e. what the user is doing), when, and where. From primary context derived context was called secondary context. For example, the identity of a person is primary context, while the phone number, the address or email addresses are secondary context.

Perera et al. [10] took this approach a bit further by redefining these two notions. They declared primary context as any information which was retrieved without using existing context (e.g. sensor readings). On the other hand they defined secondary context as any information that can be computed using primary context. This categorization was defined in terms of an operational perspective (i.e how the data was acquired) as opposed to the categorization of Abowd et al. which is a conceptual categorization (i.e what the conceptual relationships between context are). Perera et al. acknowledged location, identity, time and activity as important context types. They highlighted that in their proposed categorization scheme, the same data value can be considered as primary context in one scenario and secondary context in another scenario. As for example, location can be raw GPS data values (i.e primary context) or the name of the location like the city, a road or a restaurant (i.e secondary context).

LIFE CYCLE OF CONTEXT IN CONTEXT-AWARE SYSTEMS

A data life cycle shows how data moves from phase to phase in software systems like applications or middleware, i.e it explains where data is generated and where data is consumed. After reviewing several context life cycles, Perera et al. [10] proposed their own context life cycle , which contains a minimal number of the most essential phases. It consists of four phases (see Figure 1). Acquisition is concerned about how context is acquired from various sources. Second, the collected data need to be modeled and represented according to a meaningful manner in the Modeling phase. In the Reasoning phase, the modeled data needs to be processed to derive high-level context information from low-level context. In the last step, the context information must be distributed to the interested consumers.

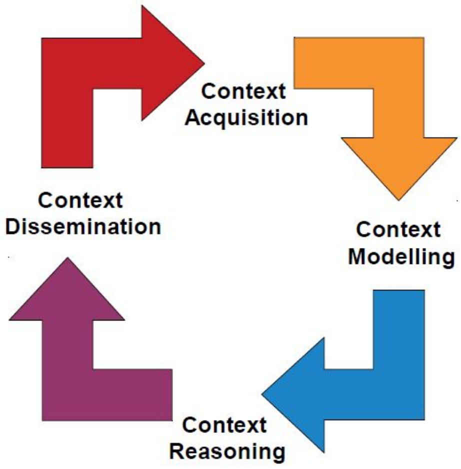

Figure 1: The context life cycle proposed by Perera et al. [10]

Context Acquisition

The process of context acquisition can be varied based on several factors. One factor is based on the frequency of events, where one distinguishes between two event types, instant events and interval events.

1. Instant events occur instantly. They are also known as threshold violations. Such events do not span across certain amounts of times, like for example the opening of a door, or a light that was switched on.

2. Interval events span a certain period of time. They are considered as periodic events, such as raining or an animal eating a plant.

Another factor is based on the type of sensors used. One can differentiate between physical sensor, virtual sensor or logical sensors.

1. Physical sensors generate data by themselves. Most of the devices we use today are equipped with a variety of sensors like temperature-, humidity-, touch-

sensors or microphones. The data retrieved from physical sensors can be considered as primary context. They are less meaningful, trivial, and vulnerable to small changes.

2. Virtual sensor do not necessarily generate data by themselves, but they retrieve data from many sources and publish it as sensor data such as calendar, contact number directory, twitter statuses, email and chat applications for example. These sensors do not have a physical presence.

3. Logical sensor, also called software sensors, combine both, physical sensors and virtual sensors in order to produce more meaningful information. A web service dedicated to provide weather information can be called a logical sensor, since weather stations use thousands of physical sensors to collect weather information but they also collect information from virtual sensor such as maps, calendars and historic data.

Context Modeling and context representation

The purpose of context modeling is to represent the previously acquired data in a meaningful manner. A well designed model is a key accessor to the context, therefore numerous approaches have been proposed to model context. context modeling typically involves two steps. First, new context information is inserted into the model and in a second step, the context is organized according to the model, i.e the context is validated and merged with existing context information. The six most popular modeling techniques are key-value models, markup schemes, graphical models, object based models, logic based models and ontology based models.

1. Key-Value Models: Context information is modeled as key-value pairs. These kind of models are simple, but it's neither possible to model hierarchical structures or relationships nor is it possible to add metadata, which makes efficient retrieval difficult.

2. Markup schemes: With markup schemes, context is modeled with tags. As such, efficient retrieval is possible, furthermore, there exist validation tools for markup scheme languages like for example XML.

3. Graphical models: Context can be modeled with relationships with graphical modeling tools like for example UML. They are richer than key-value models and markup schemes, since relationships can be modeled and its possible to use databases for fast retrieval.

4. Object based models: With object based models, context is modeled by using class hierarchies and relationships. Such models are easy to integrate into most context-aware software, since most high-level programming languages support object oriented concepts.

5. Logic based models: Context is modeled through rules and expressions. They provide more expressive richness compared to the previous model types and logic based models can supplement and enhance other types of models well.

6. Ontology based models: As the name suggests, context is modeled through ontologies. Standards exist for ontologies, for example OWL. The problem with this approach is its lack of scalability. Nonetheless, this type of model is the preferred way to model and represent context, despite its weakness.

Context Reasoning

After the acquired data has been modeled, the application has to deduce new knowledge. This phase is called context reasoning and can be divided into three steps. In the context pre-processing step, the collected sensor data is cleaned, i.e missing values are filled in, outliers are removed or the data is validated through other sources. This must be done due to inaccurate sensor readings or noise. The sensor data fusion process follows, in which the sensor data is combined from multiple sensors. This aims at producing more accurate and complete information, which would otherwise not be achieved through a single sensor. The last step is called context inference. At this stage, new high-level context information is generated be using the existing context. For this step a large number of different context reasoning decision models exist, such as decision trees, hidden Markov models, k-nearest neighbor or artificial neural networks.

Context Distribution

In the last phase, the context is distributed to the consumers. Applications, middle-ware or even end-users can be consumers. From a consumer perspective, this phase can be seen as context acquisition. However, consumers can acquire the provided context through one of two methods, either by query or by subscription.

1. Query: The consumer makes a request every time, he needs some context information.

2. Subscription: This means, the consumer first subscribes and will later on be notified, as soon as there is new context information available for distribution.

RESEARCH PROJECTS

An important role in context awareness plays physical activity, since it is considered as one of the four most important context types and the other three types like identity, time and location can be acquired quite pre-

Figure 2: The prototype of a smart glass, used in the research of Zhan et al. [4]

cise by now. Thus, a major concern is physical activity recognition. In this section, a short overview is given about three research papers, which tackle the problem of physical activity recognition through different approaches.

Ling Bao and Stephen S. Intille [2] used five small bi-axial accelerometers worn simultaneously on different parts of the body, to recognize physical activities. They claimed that prior work mostly used training data, which was collected under artificially constrained laboratory settings and thus are not representative under real-world conditions. Instead they collected data of 20 activities under semi-naturalistic conditions and were able to achieve an overall accuracy rate of 84%. The reason behind using 5 accelerometers were the advances in miniaturization which will permit accelerometers to be embedded within clothing. In their results, some miss-classifications were observed. Often, activities like "‘watching TV"’ and "‘sitting"’, or like "‘riding elevator"’ and "‘riding escalator"’ were confused.

Kai Zhan et al. [4] took another approach by categorizing daily activities into two groups in terms of the motion magnitude, namely locomotive and stationary activities. Locomotive activities are activities with high energy and specific body movements like "‘walking"’. Stationary activities on the other hand involve little or no motion such as "‘watching TV"’. Their claim was, that with the sole use of accelerometers, it was difficult to recognize stationary activities. Therefore they added visual information as an alternative method to recognize human activities. Their setup was a prototype of "‘Smart Glasses"’, a smart phone attached to the top of safety goggles (see Figure 2). They used feature extraction of both, the integrated accelerometer and the video camera and with this, were able to achieve an overall accuracy of 90% in recognizing realistic activities of daily living.

Roggen et al. [3] aimed at realizing ambient intelligence. Through standardized protocols like IPv6, the Internet of Things provides the necessary infrastructure to transparently access sensors, but still, it’s necessary to understand the user’s context, including his activity and his location. They claimed that traditional activity recognition paradigms used predefined and optimal sensor configurations and data-sets which were collected at design time. Thus, they came up with a new opportunistic activity recognition paradigm. In their novel approach, recognition methods would dynamically adapt themselves to the available sensor data. Their system adapts and expands its domain knowledge as new situations arise at run-time through self-monitoring, self-adaption, and self-learning.

CONCLUSION

Context-awareness has the potential to let applications provide more situation-specific and personalized solutions to users. To be able to build context-aware applications, we need definitions to tell whether some information is context or not. Furthermore, it’s important for designers to consider the levels of context-awareness they want to support and what types of context they want to use. A typical application passes four phases while processing context: Acquisition of sensor data, modeling of the acquired data, inference of new context information and final distribution of the newly derived information. Identity, location, time and activity are important types of context, but only the latter type is yet difficult to recognize. Thus, physical activity recognition plays an important role. The sole use of accelerometer is enough to provide accurate recognition of locomotive activities. For stationary activities, a combination of video input and accelerometers, like it may occur in smart glasses, is able to improve overall accuracy. Finally, as the Internet of Things comes up, the paradigm of such traditional physical activity recognition may change to an opportunistic one, in which the methods themselves adapt to the available sensors in the environment.

REFERENCES

1. Anind K. Dey. Understanding and Using Context. In *Personal and Ubiquitous Computing 5(1)*: 4-7 (2001)

2. Ling Bao, Stephen S. Intille. Activity Recognition from User-Annotated Acceleration Data. In *Pervasive Computing, Lecture Notes in Computer Science, Volume 3001*, pp 1-17 (2004)

3. Daniel Roggen, Gerhard Trster, Paul Lukowicz, Alois Ferscha, Jos del R. Milln, Ricardo Chavarriaga. Opportunistic Human Activity and Context Recognition. In *IEEE Computer 46(2)*: 36-45 (2013)

4. Kai Zhan, Steven Faux, Fabio Ramos. Multi-scale Conditional Random Fields for First-Person Activity Recognition. (2014)

5. Schilit B, Theimer M. Disseminating active map information to mobile hosts. In *IEEE Network* 1994

6. Schilit B, Adams N, Want R. Context-aware computing applications. In *First International Workshop on Mobile Computing Systems and Applications*, (1994)

7. Pascoe J. Adding generic contextual capabilities to wearable computers. In *Proceedings of 2nd International Symposium on Wearable Computers*, (1998)

8. Dey AK, Abowd GD. Towards a better understanding of context and context-awareness.

9. Louise Barkhuus and Anind Dey. Is Context-Aware Computing Taking Control away from the User? Three Levels of Interactivity Examined. In Proc. Ubicomp 2003

10. Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context Aware Computing for The Internet of Things: A Survey. (2014)