Thesis Work Plan
MAP-i Doctoral Programme

Mário Antunes

March 2, 2012

Supervisors: Diogo Nuno Pereira Gomes
Co-Supervisor: Rui Lus Andrade Aguiar
Thesis Title: Knowledge Extraction from unstructured Context

1 Summary

Ubiquitous and context-aware computing is an important research area wherein a lot of effort has been invested in recent years. Context-aware applications have spurred, such as location based context services and applications. Nonetheless their sophistication levels are reasonably low, as they only provide reactive services to raw context information.

It is necessary to research how to extract knowledge from multiple sources of data. One of the main problems is the definition of what is relevant information, which depends on the type of data, the source of the data, and generally speaking on the context in which it was acquired. Note that even if the relevant information is well establish for a certain scenario, the time required to derive it from the raw data can make the information useless.

It is therefore necessary to devise a system able to derive logical, useful and relevant information from the data within a time constrain related to the information itself.

2 State of the Art

2.1 Context

The most used definition of context is given by Dey and Abowd. These authors refer to context as "any information that can be used to characterize the situation of an entity (a person, place, or object) that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

According to this description, disregarding application domains, context can be practically any information as long as it is related to some entity. The excessive amplitude of this definition is pointed out by Winograd despite the value brought by the emphasis on the relationship of that information with an entity (person, place or thing). The classification of specific context is then a frequent topic in literature, probably because of this definition problem. Again, Dey and Abowd define four primary types of context evaluation: identity, activity, time, and location. These primary types of context answer the questions of "who?", "what", "when?", and "where?", and they can be used to get other sources of context.

Other popular classification method is the distinction of different context dimensions. Prekop and Burnett and Gustavsen call these dimensions external and internal, and Hofer et al. refer to it as physical and logical context. The external (physical) dimension refers to context that can be measured by hardware sensors (i.e. location, light, sound, movement, touch, temperature or air pressure) whereas the internal (logical) dimension is mostly specified by the user or captured by monitoring and inferring user interactions (i.e. the user's goals, tasks, work context, business processes, the user's emotional state).

2.2 Context Management

Pascoe, in his "wearable sensor" area identified four "core generic capabilities" in order to support context-awareness, independently of application, function, or interface:

1. contextual sensing: simple detection of environmental changes using sensors;
2. contextual adaptation: application behaviour change given these changes;
3. contextual resource discovery: discovery and exploitation of context resources;
4. contextual augmentation: associating digital data with a particular context.

While it is clear that these capabilities are somewhat generic, we can see that his definition of context is not as broad as Abowd and Dey's. In fact, Abowd, Dey and Salber notice this problem of disparate context definitions and propose a different set of requirements for a context-enabling framework:

1. Separation of Concerns: by separating how context is acquired from how it is used, a single sensor can provide context to a number of different applications, and an application can easily get different types of context information;
2. Context Interpretation: high-level socially relevant context shall be inferred from low-level physical context information, for example a meeting can be detected based on co-location, sound levels and schedules;
3. Transparent and Distributed Communications: since sensors may be physically distributed, a network is required; related to this is the need for a global time clock mechanism, so that it is possible to compare and combine context from different locations;
4. Constant Availability of Context Acquisition: because context acquiring components run independently of applications, it is not known for them when applications will require context information; consequently, these components must be always running;
5. Context Storage and History: context acquiring components should maintain a history of all obtained context; context history can be used to establish trends and predict future context values;
6. Resource Discovery: in order for an application to communicate with a context acquiring component, it must know what kind of information the component can provide, where it is located and how to communicate with it (protocol, language and mechanisms to use).

Surprisingly, these requirements, namely Separation of Concerns, Resource Discovery, Constant Availability of Context Acquisition and Transparent and Distributed Communications are much related to the problems addressed by the service orientation paradigm.

2.3 Context storage

Ubiquitous computing creates new sources of data, context-aware architectures intend to explore this increasing number of context information sources and provide richer, targeted services to end-users, while also taking into account arising privacy issues.

Before any type of context analysis can be performed, it is necessary to store and index the data. This is no easy task since the number of sources is enormous and they are continuously generating new data.

Diogo Gomes *et. al.* proposed a new context storage architecture based in Brokers (XCoA). This architecture has three main components:

1. Context Providers: entities that received and validate context data from context agents (mobile devices, social networks). Publishing in the context-broker.
2. Context Broker: entity that manages the flow of information between the context providers and context consumers.
3. Context Consumers: entities that consume information from the context-aware network.

Since all information pass through the context-broker, makes it the ideal entity to store and index the context information produced in the network.

3 Objectives

Nowadays almost every mobile device produces an enormous quantity of information. Likewise there is a emerging tendency of equipping a home with sensors that monitor several physical phenomenons. Social networks also provide enormous quantity of information about the users. Due to the quantity and the velocity by which the information is produced it is currently difficult to extract meaningful and logical conclusions from it. Context-aware architectures intend to explore this increasing number of context information sources and provide richer, targeted services to end-users, while also taking into account arising privacy issues.

The purpose of this work is to study, developed and test new techniques for on-the-fly multi-source context analysis. The main focus of this work is automatic, and if possible unsupervised, learning of logical conclusions for the purpose of detecting context situations. Identifying the context of a entity allows the system to infer logical conclusions about the entity and react based on that. The reaction can be as simple as notifying the entity about the conclusion, or as complex as automatically taking some type of action. For most real-world scenarios, this reaction has some (potentially severe) time constrains in terms of usefulness. Thus the conclusions inferred by such reasoning tool are useful only if computed in a reasonable amount of time. Various other open problems remain: what useful information can be extracted from a set of raw data? How to validate the precision of the derivative information (Quality of Context)? Those specific challenges will be also considered in the thesis.

4 Detailed description

The work shall roll-out from October 2011 to June 2014 and will be sub-divided in the following tasks:

1. Analysis of the State of the Art and Problem definition (October 2011 - January 2012): based on the current state of the art, it will be analysed what information can be extracted and how it can be used to improve the users experience while using services.

2. Formulating a hypothesis (January 2012 - June 2012): Based on the study of the state of art and the type of information that it is possible to extract relevant data is formulated a new technique. This should allow automatic learning, as much as possible unsupervised, of relevant information in a set of raw data. The definition of this technique is directly related to the study of the state of the art, and the definition of relevant information.

3. Prototype implementation (September 2012 - December 2013): After formulating a hypothesis and developing a technique that implements it, a prototype will be developed. To accomplish this step it is necessary to study currently available tools available and those that will be necessary to develop in order to mount the prototype together.

4. Evaluation (January 2013 - January 2013): During this phase the implemented prototype is evaluated and compared with alternative state of the art techniques. This experimental prototype will be explored for the practical assessment of different resources, tools and techniques.

5. Publishing results and writing Thesis (January 2013 - June 2014): Scientific articles will be written and published. During this phase the PhD thesis will be written, revised and published.