Reasoning in Context-Aware Systems

Petteri Nurmi, Patrik Floréen
Helsinki Institute for Information Technology (HIIT)
Basic Research Unit (BRU)

1 Introduction

One of the main challenges in future applications is to provide intelligent and ubiquitous/context-aware applications that take into account the user's context, e.g., the data that can be used to characterize the user's current situation. The task of using context data in an intelligent way is one of the most challenging contemporary research tasks and is often referred to as *context-reasoning*. The goal of this paper is to present the HIIT/BRU view about context-reasoning.

A more precise definition for context-reasoning is deducing new and relevant information to the use of application(s) and user(s) from the various sources of context-data. However, at this point we want to emphasize that context is by nature hierarchical in the sense that raw context-data can be further mapped into higher level categories. The raw data is called **low-level context** and the **higher level contexts** are combinations of lower level data sources. For example GPS-signals could be mapped into abstract locations such as *AT HOME*, *AT WORK* etc. Figure 1 illustrates these concepts futher.

The research problems in reasoning for context-aware applications can be approached from four main perspectives. First of all we consider the *low-level approach* that considers the research issues related to forming a view of the (user's) current context. These tasks include such as (1) *context data pre-processing*, (2) *sensor data fusion* and (3) mapping lower level context into higher level context which is also known as *context inference*. We discuss each of these in the following subsections. The second approach for context-reasoning is the so-called *application view* approach that considers the problem from the perspective of the application. This means that the

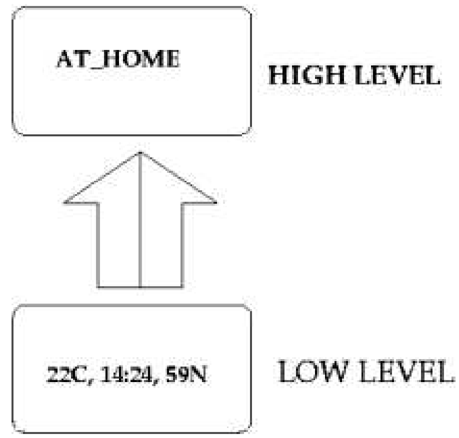

Figure 1: Illustration of the role of low and high-level contexts.

applications should be able to use the context in an intelligent way. The third approach, called *context monitoring* attempts to detect changes in the user's and application's context and to respond to the changes. Context monitoring is discussed in Section 3. Finally, the fourth approach is called *model monitoring* and it aims at keeping the learned models in a consistent state. This issue is discussed in Section 4.

2 The low-level view

The first category of problems relates to forming a view of the user's current context. We call this approach the low-level approach because most of these tasks should be supported either by the hardware (i.e. by the sensors) itself or by middleware. Some tasks probably reside in the software architectures themselves, but even then the tasks are at the lowest level of the software architecture.

The main problem that we consider in this section is the following: given the current raw context-data, how can we form the requested context snapshot, e.g. the current values of relevant context parameters, and deal with data coming from multiple sources, part of the data being possibly erroneous or missing.

2.1 Pre-processing

The pre-processing of context data aims to make later processing easier by recognizing the relevant context attributes, handling missing attributes and cleaning the data by, e.g., removing outliers. The task of recognition of relevant attributes has earlier been widely researched under the names of *dimensionality reduction* [Fod02] and *feature subset induction* [TK03] and the existing methods provide a firm basis for future applications. However, in practice the usefulness of different methods might be highly application dependent and thus it is also important to study the effects of using different pre-processing methods for context data.

Handling missing values is a very important problem in context-aware applications and thus the problem must be tackled. However, this requires real context data which must first be gathered.

The data cleaning phase is another important pre-processing phase. Many of the cleaning tasks can be seen as data mining tasks in which HIIT/BRU has a wide experience. For example, data mining algorithms [HMS01] can be used to spot outliers and to remove them.

2.2 Sensor data fusion

Sensor data fusion aims at integrating data from multiple sensors (sources) in a reliable way [LK89]. Data fusion is much researched in the field of sensor networks where the primary motivation is to reduce communication costs by the integration of similar data sources. However, in the field of context-aware computing another task of sensor (data) fusion is to recognize falsely calibrated sensors etc. For example, if there are three temperature sensors in a single room and one of them is faulty, the readings of the sensors should be combined so that the "integrated" result does not produce an outlier and is rejected in further analysis.

2.3 Context-inference

Context-inference is a very challenging task. First of all it is necessary to recognize new contexts which is discussed in the context- and model-monitoring sections. Secondly, there must be some underlying mechanism that makes it possible to map the lower level context to higher level contexts. The problem can be approached from many ways. One way is to use ontologies in the process and use logic reasoning for the mapping phase. The other approach is to use probabilistic reasoning. In the probabilistic approach variants of Bayesian networks [Nea04] are used to produce extensible probabilistic models that are then used in the mapping phase. HIIT has a wide experience in probabilistic modelling and thus it would be possible to complement the ontology approaches of other partners with a probabilistic approach.

3 Application view

The application view considers the problems after we have the context data, i.e. how to use the context data in applications. The feature subset selection / dimensionality reduction phase discussed in Section 1.1. might well be suited also for the application level. Secondly, the applications can use a wide variety of reasoning methods to use the context data. Our view in this issue is to consider reasoning components that allow the application/user to make agreements (policies) with the underlying systems on how to use the data. The components provide a common interface for similar tasks (such as classification, clustering etc.) and they can be loaded as modules into the device or can be used at an external server. In addition, such components provide a good framework for testing the performance of various reasoning methods with context data. This task is a main focus area of HIIT/BRU and thus our goal is to study which methods are useful, to implement some of them as reasoning components and to provide a framework for other applications that are to be implemented.

4 Context monitoring

The next task that we discuss is *context monitoring*. In addition to monitoring the validness of the model, which is discussed in the next section, the current context needs to be monitored. Moreover, for pro-active applications it would be useful if we were able to predict when the context is likely to change and use the prediction results as a catalyst for pro-actively performing actions such as loading software components on a mobile device. The task of context monitoring requires sequential prediction methods such as *Kalman filtering* [WB04] or *sequential Monte-Carlo sampling* (*=particle filtering*) [AMGC02] and the goal of HIIT/BRU is also to do some research on these methods — the emphasis being on particle filters.

5 Model monitoring

The model monitoring view is very important. If we learn some classifier that uses some particular contexts, it is likely that the context classes change at some point. Thus the system/application needs to recognize the changes and to update the models accordingly. Also the decisions made by the applications should be monitored and user feedback should be used to modify the behaviour of the system. Learning from user feedback is better known as *reinforcement learning* in the machine learning community.

Recognition of changes can be done, e.g., using time-series analysis [Ham94] and (non-hierarchical) clustering methods [JMF99]. Researching methods for model monitoring as well as researching how to use reinforcement learning methods [KLM96] to adapt the behaviour of the systems/applications is a main task of HIIT/BRU. We would like to emphasize that in reinforcement learning user feedback need not be explicitely asked for (i.e. the user needs not to press yes or no), but a more subtle approach to get the user feedback can be used: e.g., if the user is not satisified with the choice made by the application, (s)he is more likely to return to the previous application view or to switch to a completely new one. Registering this can be interpreted as a negative feedback for the system and reinforcement learning methods can be used to modify the performance accordingly.

References

[AMGC02] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174 – 188, 2002.

[Fod02] Imola K. Fodor. A survey of dimensionality reduction techniques. Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, Livermore, CA 94551, jun 2002.

[Ham94] James D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, NJ, 1994.

[HMS01] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. MIT Press, Cambridge, MA, 2001.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[KLM96] Leslie P. Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Artificial Intelligence Research*, (4):237–285, 1996.

[LK89] Ren C. Luo and Michael G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19:901 – 931, sept-oct 1989.

[Nea04] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, NJ, 2004.

[TK03] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Academic Press, London, UK, 2nd edition, 2003.

[WB04] G. Welch and G. Bishop. An introduction to the Kalman filter. TR95 -041, University of North Carolina, 2004.