AN ARCHITECTURE FOR CONTEXT PREDICTION

Rene Mayrhofer*

Abstract

Today's information appliances appear very powerful, featuring on-device storage and processing power, communication technology and supporting many different applications. Context awareness is currently considered as one of the key issues for future device generations, with context prediction being the next step in research. The goal is not only to recognize the current context of an information appliance or its user, but also to predict the future context and thus enable the device to become proactive. In this paper, an approach to recognize and predict high level context information from low level sensor data is presented. Targeting a wide range of platforms, this approach has also been implemented in a software framework for on-line, un-supervised context prediction.

1. Introduction

Information appliances are either mobile, like laptop computers, handheld devices, mobile phones or wearables, or fixed, like TV set-top boxes, home entertainment centers or even whole rooms equipped with various interacting devices; but most of them have various hardware components that can be used as sensors for querying the environment. By exploiting these sensors, it is possible to make devices context aware and thus adaptive to the user's current situation. This paper presents the basic methodology and its implementation as a software framework which eases the development of context aware applications by providing the current and future, predicted context.

Context awareness allows an information appliance to adapt to its environment, which offers a number of advantages and possibilities for new applications. One of the more intuitive advantages is ease of use; if devices can adapt to their – or for personal, mobile information appliances their user's – situation, they can engage in more efficient user interaction (e.g. [2, 10]). Other advantages may be less obvious, but context awareness can also lead to reduced energy consumption (cf. [20]) and thus, for mobile devices, to longer battery life (e.g. [16]). Another option for improving the usability of information appliances [21] is to make them proactive: anticipating user action enables a new class of applications to be developed. The main contribution of this work is on-line, adaptive prediction of device and/or user context. This can be seen as a special case of context awareness, which considers the past, present or future context of an appliance. This work concentrates on providing context information to applications. It is left to the applications what the context information is used for; in many cases, simply displaying the current context and adapting the user interface to it might be a better choice than automatically starting certain actions.

*Institut für Pervasive Computing, Johannes Kepler Universität Linz, Altenbergerstr. 69, 4040 Linz, Austria, rene@soft.uni-linz.ac.at

Some of the issues involved with fulfilling the above aims are already solved by different methods and techniques. It is not necessary to develop new algorithms for classification of sensor data or prediction of time series, since mature methods are already available. The major challenges for this research are that:

- Context recognition and prediction should be embedded in information appliances with limited resources. This limits the set of suitable algorithms significantly.
- Learning and adaption should happen on-line without explicit training, which also excludes a number of powerful algorithms that rely on batch training.
- User interaction should be kept to a minimum and be un-obtrusive.
- Values gathered from typically available sensors are highly heterogeneous and thus many algorithms for statistical analysis and classification are not directly applicable.
- Prediction should respect trends and periodic patterns in context history with a reasonable trade-off between forecast accuracy and computational complexity.

To accomplish this, the presented architecture consists of four major parts: feature extraction, classification, labeling and prediction. The available sensors provide a multi-dimensional, highly heterogeneous input vector as input to the classification step, realized by data clustering. Labeling associates recognized context classes with meaningful names specified by the user, and prediction allows to forecast future user context for proactive behavior.

To facilitate the development of applications that build upon predicted context, one aim is to develop a framework for recognizing and predicting context from a multitude of simple sensors available on current information appliances. Specifically, the aims of this framework are:

- Provide abstract, high level context information to applications
- Use multiple, simple, available sensors in favor of additional, more powerful ones that would need to be connected externally
- On-line, un-supervised context recognition and prediction without explicit training phases
- Local operation (i.e. no infrastructure necessary)
- Unobtrusive operation with explicit user interaction only when necessary; context awareness should aid the user, but not cause additional burden
- Cope with limited resources of typical mobile devices (e.g. mobile phones)
- Run on a variety of platforms and operating systems and independently of the specific sensor technology

The developed architecture is a general approach to context prediction and thus suitable for a wide range of applications. Domain-specific heuristics might achieve higher accuracy for special applications, but the strengths of this architecture lie in its flexibility. Thus, it is probably better suited for applications utilizing many different sensors than for applications with a single sensor and well-known, highly optimized, domain-specific methods.

This paper is structured as follows: section 2 discusses related work and puts this work in context. Section 3 describes the general approach, the developed framework and the major contributions. Finally, section 4 gives concluding remarks and future outlook.

2. Related Work

Context awareness is currently a highly active research topic [4], but most publications assume few but powerful sensors like video or infrastructure based location-tracking. The TEA [25] and Smart-Its [11] projects have proposed to use multiple simple sensors instead of a single, powerful one to better capture the different aspects of context [24]. Our work takes a comparable approach to context detection by using multiple diverse sensors, but extends it to also exploit qualitative, non-numerical features [18]; many of the currently available sensors produce non-numerical values, like the list of Bluetooth-enabled devices in range. Although such sensor data can not easily be mapped to numerical features, they offer valuable information about the user or device context and should thus be exploited. The developed framework allows the transparent integration of features of arbitrary types and has been presented in [19].

The notion of feature extraction has also been used in the field of context awareness in [5, 6, 13, 17]. Feature extraction and sensor fusion have also been studied extensively in the field of robotics, but with focus on geometrical properties of the environment (e.g. [3]). In the field of context awareness, sensor fusion is not really applied because the different sensors capture different aspects of the environment. Sharing of context information between devices in spatial proximity has also been proposed independently in [17], similar to our recommendation in [19].

In [9], Anind K. Dey et.al. described a software infrastructure for context awareness, which depends on a server for aggregating context and is limited to discrete-valued types of sensors. This infrastructure has been implemented and named Context Toolkit [23] with its main abstractions of ”widgets”, ”aggregators” and ”interpreters” [8]. It is not directly comparable to our work because of different aims; we aim to implement context recognition and prediction locally on each device, without the need for infrastructure components. Another interesting toolkit is iStuff [1], which concentrates on explicit user interaction; we specifically aim for non-obtrusiveness and almost no explicit user interaction for recognizing and predicting context. Due to their complementary nature, both approaches could be integrated into applications on top of the frameworks. Additionally, data obtained from the explicit user interaction supported by iStuff could be used as (logical) sensors for our approach.

Proactive adaptation of applications has also been explored in [14], which describes CIS, the “Contextual Information Service” as a lightweight interface to obtaining context information. Their notion of “Contextual Information Providers” can be compared to feature extractors in our work, described shortly in section 3.2., or to “cues” in the TEA project. However, CIS follows the approach of adapting the environment (i.e. infrastructure), while we intend to adapt the device to a changing environment, which includes changing user behavior. The use of meta-attributes in CIS at the level of context primitives seems to be an interesting concept; we currently use confidence as a meta-information only at higher levels (after classification and after prediction).

Learning user’s habits has previously been explored by Michael C. Mozer in The Neural Network

House [20], which is able to predict occupancy of rooms, hot water usage and likelihood that a zone is entered in the next few seconds using trained feed-forward neural networks. Kidd et.al. reported [15] about the Aware House, which should also learn user's habits, but was not finished at the time of the report. A more recent project called MavHome [7] by Diana J. Cook et.al. also applies prediction algorithms to forecast future user actions, but parts of the prediction seem to rely on database support and batch training.

3. Framework

The major aim of this work is to develop an architecture for recognizing and predicting context, which includes the methodical evaluation and selection of algorithms for different parts of the architecture. In this paper, the general approach and current selection of algorithms is presented. However, one of the most important features of the developed architecture is its flexibility and independence of any specific algorithms for context classification and prediction. On most supported platforms, the algorithms can even be selected during run-time due to a plug-in concept. To evaluate the architecture and its capability to fulfill the stated aims, it has been implemented as a software framework.

3.1. Motivation

After an extensive literature study, a software framework for the development of context-aware applications on resource limited devices without any infrastructure support does not seem to be available. Instead, there are either powerful frameworks for collecting context information from multiple devices over some network infrastructure (e.g. [7, 9, 11, 14, 22]) or ad-hoc applications tailored to a specific hardware platform (e.g. [5, 6, 10, 16, 20, 25]).

The current rise of information appliances like the Symbian OS based smart phones or standard PDAs motivates the development of a cross-platform software framework for exploiting all available sensors to locally recognize and predict context. Although the framework should not depend on infrastructure components so as to be usable on mobile and disconnected devices, it should be possible to integrate infrastructure services as additional sources of information about the environment. Examples are a network camera, WLAN information from an access point or a central logon server; all of them can contribute valuable aspects of the current context, depending on the application.

3.2. Approach

Figure 1 shows the general architecture of the framework. As can be seen, there are four clearly distinguished steps separated by simple interfaces. Due to the exact definition of the interfaces between the steps, they are mostly independent and can be exchanged independently. All of the feature extractors, the classification algorithm and the prediction algorithms have been realized as dynamically loadable shared libraries which can be selected during run-time.

In its current state, the architecture has been optimized towards mobile and embedded devices, typically with restricted GUI capabilities. The different processing steps can be regarded as filters, transforming input values to output values. No central database is necessary, every step is independent and performs on-line data processing. Thus, the architecture is well suited for resource limited information appliances, but does not impose a boundary on the complexity of specific plug-ins. When applied to a pervasive computing environment like a room or a whole house, powerful computing infrastructure can be utilized. Although plug-ins from different steps, e.g. a classifier and a predictor,

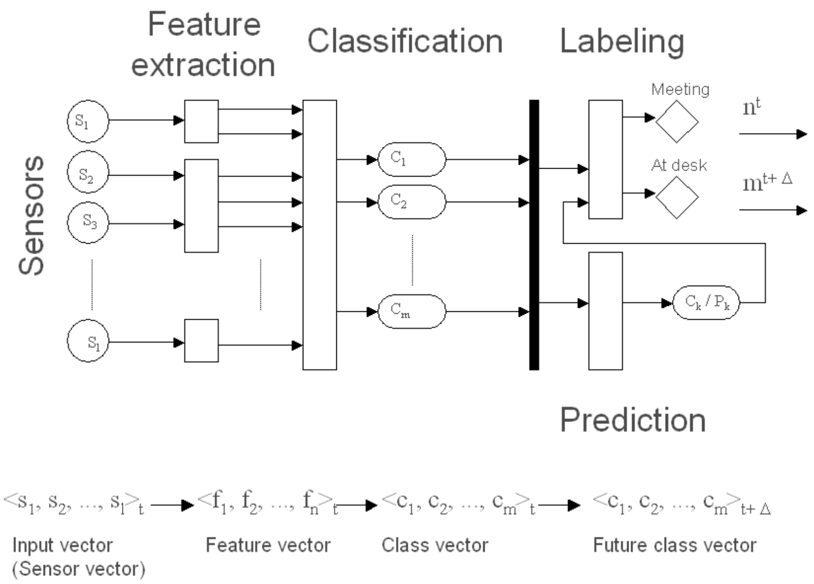

Figure 1. Architecture for proactivity via predicted user contexts

might use extensive data storage facilities like a central database server, they can do so independently. The interfaces between the distinctive steps are vectors attributed with time stamps. Thus, time is implicitly passed from one step to the next. For debugging, evaluation and monitoring purposes, the complete history of these vectors can be logged in a general way and thus allows off-line processing if necessary. During normal operation in an information appliance, no central history is necessary as each plug-in maintains it own, internal data model. Typically, this will not cause any duplication of data because the plug-ins in different steps will construct data models with different meanings.

The main advantages of this architecture, w.r.t. to the aims defined in section 1 are that it allows an on-line operation without explicit training phases, that it is non-obtrusive due to the use of unsupervised methods and that it is able to cope with limited resources because no database is required. Furthermore, the plug-in architecture allows classification and prediction algorithms to be exchanged easily, which simplifies the evaluation phase for specific applications. Evaluation of the architecture will be done through a series of experiments with real world data. This will allow a qualitative assessment if the whole system performs as expected. A quantitative evaluation can not be done for the architecture per se, but only for a selection of specific methods realized as plug-ins.

Currently, Growing Neural Gas is used as a classification algorithm, because it offers a number of advantages in terms of adaptivity. To further improve its result quality when working with highly heterogeneous feature vectors, a few enhancements have been developed and will be presented in more detail in future publications. A first evaluation of prediction methods showed that Active LeZi [12] is a good candidate, at least for single-step context prediction. On a simple, artificial test data set, a prediction accuracy of 0.99 can already be attained for a single feature space dimension; simple ARMA models did not yield any usable results on that data set without problem specific modeling and manual selection of input values, while Active LeZi constructs its internal model in an un-supervised way. For real-world data, quantitative evaluation still needs to be done. As a first prototype for the labeling step, a simple traybar application displays the currently active context and allows the user to interactively assign descriptive names to context identifiers.

3.3. Contribution

The main contribution of this work can be seen in two interrelated areas:

- The development of an architecture for context prediction which features:
  - flexibility due to simple interfaces between multiple well-defined steps. This makes the architecture suitable for arbitrary kinds of sensors and a wide range of appliances. *and*
  - clear separation of concerns. Sensors, feature extractors, classifiers, predictors and user interaction are completely independent and interchangeable.
- The implementation of the architecture with the following novelties:
  - Use of heterogeneous feature vectors for context recognition: Although the problem of heterogeneity has been investigated for input values of Kohonen Self Organizing Maps, publications on context recognition only seem to concentrate on numerical, mostly continuous feature types (e.g. accelerometer data). This work introduces a method of using a broader class of features for deriving context.
  - Context prediction: Some publications already covered the topic of location prediction for different granularities of location information; most projects perform prediction of user location on the level of rooms. The concept of predicting the whole context on the level of abstract context identifiers with on-line algorithms locally at an information appliances seems to be novel.
  - Development of a cross-platform framework suitable for devices with limited resources: As outlined in section 3.1, no framework seems to be available which is suitable for fulfilling our aims.

Because it needs to cope with limited resources and access many hardware components, the framework has been implemented in ANSI C++ with resource constraints in mind. Although C++ is theoretically not as portable as other languages like Java, practical experience with both shows that Java programs also need modifications and sometimes even structural redesign when being ported to different platforms (e.g MIDP on mobile phones). Currently, the framework and most of its supported sensors run on Linux (IA-32 and ARM processors), Windows 2000 and XP, Windows CE 3.0 and, with restrictions, on Symbian OS 7.0. It is freely available under an open-source license upon email request.

3.4. Open Issues

At the moment, the conceptual work, basic framework structure and feature extractors for a variety of sensors are complete. An enhancement of Lifelong Growing Neural Gas (LLGNG) has been implemented for the classification step and tested with artificial and real world data. However, further optimizations of the LLGNG parameters might still increase the classification quality and will therefore be investigated. In addition to the qualitative comparison of different classification methods which has already been done, a quantitative evaluation of one or two additional algorithms could provide valuable insight; but it is difficult to find on-line classification algorithms that fulfill most of the requirements we have defined [19]. The major aim of this work is to develop and evaluate an architecture for context prediction, the evaluation of specific algorithms is only necessary for validating the architecture as a whole. For the prediction step, only the Active LeZi algorithm has been implemented within the framework so far. Other prediction methods like ARMA models have been tested in Matlab with the data sets produced by the classification step of the framework, but suggest further research. More complex, non-linear models still have to be evaluated on this test data. An online statistical analysis of the time series will be implemented within the framework to provide hints to prediction algorithms. This could also allow to efficiently apply multiple algorithms and combine their results for an increased overall prediction accuracy.

4. Conclusions and Outlook

Two contributions to context recognition and prediction can be identified in this work and have been shortly presented: a flexible multi-step architecture for context recognition and prediction has been developed and has been implemented in form of a software framework. This framework has been developed for a wide range of platforms and with support for many common, currently available sensors. In its current form, it can already be regarded as a mature infrastructure for gathering and pre-processing sensor data and will be used for multiple currently active research projects within our institute.

Acknowledgments

The presented architecture has been developed together with my supervisor, Prof. Alois Ferscha, and Harald Radi. Most implementations have been written in close cooperation with Harald Radi and would probably not exist in their current form without his excellent contributions.

References

[1] R. Ballagas, M. Ringel, M. Stone, and J. Borchers. iStuff: A physical user interface toolkit for ubiquitous computing environments. In *Proceedings of the ACM CHI 2003 Conference on Human Factors in Computing Systems*, pages 537–544, April 2003.

[2] M. Beigl, H.-W. Gellersen, and A. Schmidt. Mediacups: experience with design and use of computer-augmented everyday artifacts. *Computer Networks (Amsterdam, Netherlands: 1999)*, 35(4):401–409, 2001.

[3] S. B. H. Bruder. An information centric approach to heterogeneous multi-sensor integration for robotic applications. *Robotics and Autonomous Systems*, 26(4), March 1999.

[4] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000.

[5] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. Technical Report 471, MIT Media Lab, Perceptual Computing Group, 1998.

[6] B. Clarkson, N. Sawhney, and A.Pentland. Auditory context awareness via wearable computing. In *Proceedings of the 1998 Workshop on Perceptual User Interfaces (PUI'98)*, San Francisco, CA, USA, November 1998.

[7] D. J. Cook, M. Youngblood, III E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. MavHome: An agent-based smart home. In *First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pages 521–524. IEEE Computer Society Press, March 2003.

[8] A. K. Dey and G. D. Abowd. The context toolkit: Aiding the development of context-aware applications. In *Workshop on Software Engineering for Wearable and Pervasive Computing*, June 2000.

[9] A. K. Dey, G. D. Abowd, and D. Salber. A context-based infrastructure for smart environments. In *Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99)*, pages 114–128, 1999.

[10] A. K. Dey, D. Salber, G. D. Abowd, and M. Futakawa. The conference assistant: Combining context-awareness with wearable computing. In *ISWC*, pages 21–28, 1999.

[11] H.W. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artefacts. *Accepted for publication in Mobile Networks and Applications*, 2003.

[12] K. Gopalratnam and D. J. Cook. Active Lezi: An incremental parsing algorithm for sequential prediction. In *Proceedings of the Florida Artificial Intelligence Research Symposium*, 2003.

[13] R. Headon. Movement awareness for a sentient environment. In *First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pages 99–106. IEEE Computer Society Press, March 2003.

[14] G. Judd and P. Steenkiste. Providing contextual information to pervasive computing applications. In *First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pages 133–142. IEEE Computer Society Press, March 2003.

[15] C. D. Kidd, R. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. D. Mynatt, T. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proceedings of the Cooperative Buildings, Integrating Information, Organization, and Architecture, Second International Workshop, CoBuild'99*, volume 1670 of *Lecture Notes in Computer Science*, pages 191–198. Springer, 1999.

[16] S. Loughran. Towards an adaptive and context aware laptop. Technical Report HPL-2001-158, Hewlett-Packard Labs, June 2001.

[17] Jani Mäntyjärvi, Johan Himberg, and Pertti Huuskonen. Collaborative context recognition for handheld devices. In *First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pages 161–168. IEEE Computer Society Press, March 2003.

[18] R. Mayrhofer, H. Radi, and A. Ferscha. Feature extraction in wireless personal and local area networks. In Khaldoun Al Agha and Cambyse Guy Omidyar, editors, *The Proceedings of The Fifth IFIP-TC6 International Conference on Mobile and Wireless Communications Networks (MWCN 2003)*, pages 195–198. World Scientific, October 2003.

[19] R. Mayrhofer, H. Radi, and A. Ferscha. Recognizing and predicting context by learning from user behavior. In W. Schreiner G. Kotsis, A. Ferscha and K. Ibrahim, editors, *The International Conference On Advances in Mobile Multimedia (MoMM2003)*, volume 171, pages 25–35. Austrian Computer Society (OCG), September 2003.

[20] M. C. Mozer. The neural network house: An environment that adapts to its inhabitants. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pages 110–114. AAAI Press, 1998.

[21] D. A. Norman. *The Invisible Computer*. MIT Press, Cambridge, 1998.

[22] Deliverable D05: 1st year progress report of the Oresteia project. Technical report, January 2002. available at http://manolito.image.ece.ntua.gr/oresteia/htmldocs/deliverables.htm.

[23] D. Salber, A. K. Dey, and G. D. Abowd. he context toolkit: Aiding the development of context-enabled applications. In *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99)*, pages 434–441, May 1999.

[24] A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.

[25] A. Schmidt and K. Van Laerhoven. How to build smart appliances. *IEEE Personal Communications*, August 2001:66–71, 2001.