Context Prediction based on Context Histories: Expected Benefits, Issues and Current State-of-the-Art

Rene Mayrhofer
Johannes Kepler University Linz
Altenbergerstr. 69
4040 Linz, Austria
+43 732 2468 8527
rene@soft.uni-linz.ac.at

ABSTRACT

This paper presents the topic of context prediction as one possibility to exploit context histories. It lists some expected benefits of context prediction for certain application areas and discusses the associated issues in terms of accuracy, fault tolerance, unobtrusive operation, user acceptance, problem complexity and privacy. After identifying the challenges in context prediction, a first approach is summarized briefly. This approach, when applied to recorded context histories, builds upon three steps of a previously introduced software architecture: feature extraction, classification and prediction. Open issues remain in the areas of prediction accuracy, dealing with limited resources, sharing of context information and user studies.

Keywords

Context prediction, context histories, time series prediction, machine learning

INTRODUCTION

Context histories, especially when recorded over a long term, offer a wide range of possibilities to enhance the services provided by some computer system. These possibilities include inferring of current and past user actions, selection of devices, etc. However, the prediction of future context based on the recorded past contexts is often conceived as the ultimate challenge in exploiting context histories. Context prediction, i.e. exploiting expected future context, can offer distinct advantages over the sole usage of past and current contexts: Obviously, it could be used to perform actions on behalf of the user, but this is problematic and will be discussed in more detail. However, it is also possible to exploit predicted context even without triggering actions in the physical world. On the one hand, comparing predicted contexts with recognized ones allows to detect irregularities and therefore assists in dealing with system failures. On the other hand, proactivity allows to provide user interaction that conforms better to the user's expectations. The efficiency of interpersonal communication builds upon a shared understanding of the past, current and last but not least future context within which interactions take place. Computer systems usually do not share such an understanding, and therefore at least a partial awareness of the relevant contexts is a prerequisite for a significant improvement of user interaction.

In addition to improving the human/computer interaction that is needed for most application areas, the introduction of proactivity opens new possibilities for automating application areas that are discussed in more detail in the next section. There are many examples from different areas that can benefit from an integration of proactivity: e.g. traffic and logistics (continuous planning and adaptation building upon estimated times of arrival, optimal utilization of road and parking place capacities, prevention of traffic jams), manufacturing (detection of and dealing with exceptions in just-in-time processes, planning for flexible manufacturing systems), individual traffic (prediction of arrival by the vehicle, warning before traffic jams, initializing or booting on board systems before they are used to prevent delays), medical care (alerting or initiating counter measures before critical situations can occur, digital dietary assistants that are aware of personal habits and predicted future events), communication (in-time establishment or change of connections, improved roaming, data synchronization and controlled shut down of sessions before connections are terminated), home automation (in-time establishment of custom room temperatures, reordering of groceries or fuel), etc. In combination with context awareness, proactivity opens numerous possibilities to enhance available informational services or construct new, currently unavailable ones. The next section presents a first taxonomy of applications that can benefit from context prediction.

Complementing the potentially large benefits of context prediction, there are serious issues with its technical, social and last but not least legal aspects. An overview of the currently perceived most important issues is given in a separate section. After listing the current issues, a first approach to context prediction based on recorded context histories is given and the remaining open issues are listed.

POTENTIAL BENEFITS OF CONTEXT PREDICTION

Although context prediction can be useful in most applications that currently utilize context awareness in general, a few application areas have been identified that benefit significantly from the introduction of proactivity. These application areas all focus on one central maxim: *to avoid potential problems caused by erroneous predictions*. As already mentioned in the introduction, an obvious benefit of context prediction is that it enables systems to perform actions on behalf of the user, like booking flight tickets when a potentially interesting conference will be held or ordering groceries when friends are invited to dinner. These two examples already indicate that automatically triggering actions based on context predictions is a delicate issue. What if the conference is indeed very interesting to the user and the system thus determines that she will attend it, but the budget does not allow for it? What if the invited group of friends decides spontaneously to go to a nearby restaurant instead? Predictions of future events will necessarily be imprecise, and in some cases they might even be impossible (cf. [5]). Therefore, we strongly suggest that systems that exploit context prediction should impose a design principle of not automatically triggering actions that can cause serious real world effects whenever a prediction is uncertain. Although the following areas of reconfiguration and accident prevention might influence real world objects, the effects of erroneous predictions tend to be limited. The following taxonomy of application areas that lend themselves to context prediction at the current state of research has first been presented in [4] and is summarized here:

Reconfiguration

System reconfiguration in general, not being restricted to context-based reconfiguration, is today one of the most time-consuming tasks associated with computer systems. We can further distinguish between *light-weight* and *heavy-weight reconfiguration*, where light-weight reconfiguration includes modification of the system configuration or general online, near real time adaptation to changed environments. Heavy-weight reconfiguration includes tasks that impose a noticeable delay during reconfiguration, leaving the system in question out of service until reconfiguration has finished. Boot-up of systems, installation or update of applications, maintenance and infrastructural changes, downloads, searches in large databases, etc. all consume, or even waste, significant amounts of valuable work-time. Any progress towards shortening these reconfiguration times yields a direct improvement for the involved people. We believe that such heavy-weight reconfiguration can be performed in advance by exploiting context histories to predict future context.

Accident prevention

An accident can be seen in the general case as an *undesirable system state*, and preventing such undesirable states has applications in many different areas. E.g. in telecommunication, an undesirable state is an overload in some network equipment or communication link. Load prediction is already used by larger telecommunication organizations to prevent system failures by proactively updating or bypassing highly loaded systems in time. In medical care, there is a vast multitude of undesirable or dangerous states and situations than can be monitored with bio sensors and should be predicted to prevent permanent damage. However, it is important to point out that with the approach presented in [4], such an undesirable situation must have already occurred in the past to be predicted. Lacking an application-specific model of desirable and undesirable situations – which can not be assumed when we focus on exploiting context histories – it is only possible to learn from past situations. Yet unknown contexts can not be predicted in a general, application-independent way. Thus, we strongly advise against using this approach for prediction and prevention of undesirable states in safety-critical systems.

Alerting

This is best known from the domain of PIM (personal information manager) type applications, including calendar, project management, scheduling, appointment and group coordination systems, but includes arbitrary applications that need to alert users in some form. These systems already provide a multitude of alerting capabilities, ranging from message boxes bound to being displayed on desktop computers or PDAs, signal lights, audible notification to sending emails, SMS or pager messages to user's mobile devices. However, events leading to such alerts are either triggered by certain actions (e.g. a colleague entering a virtual meeting room) or have been scheduled in advance, being entered in a calendar. When being able to predict future context, a device can autonomously issue alerts before some relevant contexts occur, without the need for manual scheduling.

Planning aid

Simply displaying predicted future context in a structured way and allowing to interactively browse it can provide a powerful aid for human-driven planning and scheduling. This puts people in the control loop, allowing to manually modify system behavior, but being assisted by predictions of future situations. Due to their informational nature, applications from this area will demand an estimation on the probability of the predictions being true, i.e. on their certainty, which might not be strictly necessary for other application areas.

In these application areas, context prediction can be exploited to provide better services to the user, but the effects of erroneous predictions should still be easy to undo or be even unnoticeable to the user because they can be reverted automatically. Until the certainty of predictions can be estimated satisfactorily to decide which predictions can be trusted and which can't, this is an important feature. The following two sections present the aspects that need to be considered for context prediction, i.e. for all of the discussed application areas, as well as issues that appear when actually building such systems.

ASPECTS OF CONTEXT PREDICTION

There are many different aspects that need to be considered for context prediction, as it involves the recording of context histories, context recognition, time series prediction and acting on the real world based on those predictions. Initial experience with designing and implementing proactive systems shows that the following aspects are among the most important:

Time Series Aspects

The prediction engine should consider *sequential patterns*, *periodic patterns*, *long term trends* and possibly also *exceptions*. To enable the unsupervised recognition of periodic patterns, the length of the context history must include at least a few cycles of the longest period that should be detected. E.g., if seasonal effects should be predicted, a few years will need to be recorded continuously.

Training Aspects

Another important aspect is the kind of training, i.e. how the model is constructed. In machine learning, classification and prediction methods are usually distinguished as *supervised*, i.e. the target values are known for the training set, and *unsupervised*, i.e. only the input values are known.

This distinction should not be confused with the involvement of human experts in the training process, which is an orthogonal classification of approaches; such data mining approaches are sometimes also denoted as supervised approaches, although the involvement of users in the model construction is possible for supervised and unsupervised methods. For exploiting context histories, all options need to be considered, i.e. if the model is constructed automatically or via an interactive process involving human experts and if the approach is supervised or unsupervised. The associated issues are shortly discussed in the next section, but the appropriate selection of the training method is typically highly application dependent.

Context History Aspects

Additional aspects evolve rather around the recording of context histories than the usage of these histories for context prediction, but influence their exploitation and are consequently also discussed here. One of the most important aspects is the acquisition of ground truth, i.e. if "true" output values like user-specified context identifiers are recorded alongside the raw sensor data or not. Without such a ground truth, a quantitative assessment of the results is difficult, and often impossible (cf. [4]).

Another important aspect is the location of context histories, i.e. if they are stored in a centralized or a decentralized way. With decentralized storage, different costs of accessing other parts of the history arise and need to be considered in the usage of these histories. An example is the storage of short term histories locally at the system involved in the context prediction and long term histories on mass storage devices. Accessing long term history allows detecting periodical patterns of longer period, but involves higher cost.

The third aspect concerning the acquisition and recording of context histories, which is partially interrelated with the location, is the level of data that is recorded. There is a wide range of possibilities for context data on different levels, ranging from raw, unprocessed sensor data via data on the feature level to pre-classified context identifiers (cf. [4]). The higher the level of the recorded data, the less data typically needs to be stored, but the higher the effort for acquisition. Resource limited devices like nodes of a sensor network might even be incapable of the necessary pre-processing for recording data on any level other than raw sensor data. For context prediction, completely different approaches are necessary depending on the level of context data.

ISSUES

This section discusses issues of context prediction that emerged in most recent research on that topic and, for some of them, potential solutions or recommendations w.r.t. context histories. The following issues are likely to be present in nearly arbitrary uses of context histories:

- *Accuracy*: The accuracy of the recorded data is the factor with the highest influence w.r.t. the result quality. From the aims of a context history, the required accuracy and consequently the necessary sensor technology can be deduced.
- *Fault tolerance*: In real world experiments, missing values due to failing sensors and the inherent noise in sensor time series need to be dealt with. A more complex case are erroneous sensors that can not be detected directly as failing – which would allow to record missing values for the specific sensors – but that yield biased or completely erroneous values. These are more difficult to deal with than the "no-value" failures or the usual noise and often need sensor-level redundancy to compensate.
- *Unobtrusive operation*: Recording long-term histories, which are necessary for learning user behavior from scratch, i.e. without expert knowledge, or recording data from multiple users requires an unobtrusive operation that does not interfere with the normal activities of the test subjects. This is necessary both for the recording and for the usage of context histories in practical applications.
- *User acceptance*: In every exploitation of context histories and often even in their recording, a felt loss of control of involved users is a serious problem that currently needs to be addressed in an application specific way, which might include non-technical means like user education or organizational changes.
- *Privacy*: Closely related to the previous issue is the area of privacy – including legal aspects that are still to be clarified. Privacy issues might also lead to problems with user acceptance, but typically only few users are aware of the implications of recording extensive context histories.

Therefore, privacy issues must be tackled by the designers of experiments and systems that record context histories.

More specific to context prediction are the following issues:

- *Supervised vs. unsupervised*: Only when ground truth is available, supervised learning methods can be used. With unsupervised methods, evaluation of results is more difficult. For context prediction, ground truth for training purposes can be extracted from any recorded context trajectory, i.e. context time series, by splitting the trajectory into a training and a test set. For evaluating the prediction results, the test set can be used. When context prediction is used in online systems, ground truth can not be known immediately, but becomes available when the predicted time has passed.
- *Automatic vs. manually assisted*: For small data sets, human experts can construct the respective prediction model, possibly assisted by data mining techniques or suggestions by the system. This expert-driven approach is only feasible for few experiments, but usually not for the independent prediction of the contexts of many users. In this case, the prediction model needs to be constructed automatically, based solely on the available context history and potentially some domain specific knowledge that has been embedded into the learning process.
- *Problem complexity*: A serious issue is the general complexity of time series prediction problems w.r.t. the size of the recorded data sets and run-time complexity for constructing the models and subsequently determining predictions based on the models. Most of the more powerful prediction techniques bring forth considerable demands for processing and storage capabilities and might thus be unsuitable for embedded or mobile systems.
- *Uncertainty*: Using uncertain predictions to act on the real world is generally problematic, as discussed in more detail above; if there is any doubt about some prediction of future context – and in almost any cases doubt is expedient in time series prediction – then it is recommended to “play safe” and not to depend on the predictions for critical actions. It is generally advisable to leave the user in the control loop (cf. [1]).
- *Online processing*: If context prediction – or in fact any exploitation of context histories – should be embedded into computer systems in the spirit of pervasive computing, it needs to happen online, without a distinction into training and usage phases. A device must be continuously available and must be adaptive to changing environments. This makes it impossible to use some learning methods that depend on batch training.
- *Heterogeneity*: Values gathered from typically available sensors are highly heterogeneous and thus many algorithms for statistical analysis and classification are not directly applicable.

CURRENT APPROACHES

The current approach to context prediction suggested in [4] is the prediction of abstract contexts – in contrast to the autonomous prediction of individual aspects like the geographical position of the user. It is based on a multi-step software architecture that separates context recognition, i.e. the classification of raw sensor data to higher-level context identifiers, from context prediction, which is based on the trajectories of context identifiers. This approach is inherently decentralized, because each device is supposed to recognize and predict context independently. By distributing the acquisition and exploitation of context histories, privacy issues are mostly avoided, as long as the personal device that records and predicts context is trusted. Privacy and the issue of limited resources are also supported by the use of online methods as far as possible. This way, only a sliding window of the context history needs to be available instead of the complete time series data. [4] gives an overview of prediction methods suitable for context prediction. Particular attention is turned to implicit user interaction to prevent disruptions of users during their normal tasks and to continuous adaptation of the developed systems to changing conditions. Another considered aspect is the economical use of resources to allow the integration of context prediction into embedded systems. The developed architecture has been implemented as a flexible software framework and evaluated with recorded real-world data from everyday situations.

Other approaches that have not yet been studied in detail by the author are to predict sensor or feature level data instead of context identifiers, and to record and use the complete context history. The former has the advantage that domain-specific knowledge about the sensors can be exploited for prediction, e.g. geographical maps for location prediction, but the disadvantage that correlations between different sensors are not considered in the prediction model. This is also an issue for many other uses of context histories, since the interrelations between different sensor time series are often not apparent at the lower levels of raw sensor data but need to be recognized by applications using those histories. The latter approach allows the usage of more powerful prediction methods, but imposes significantly higher demands on the storage capabilities of involved devices, which is again not limited to context prediction but is true for arbitrary uses of context histories.

CONTRIBUTION

The present position paper discusses potential benefits, aspects and current issues of context prediction and briefly summarizes a first approach as presented in more detail in [4]. This approach considers – and partially addresses – the issues of fault tolerance, unobtrusive operation, privacy, unsupervised context recognition, automatic construction of the prediction model, online processing, and heterogeneity. The issue of uncertainty is shifted to applications implemented on top of the developed architecture, but dealing with it at application level is assisted by providing measures of certainty of recognized and predicted contexts. Both major parts of this architecture, i.e. the recognition and the prediction parts, can be used independently and thus allow to record and to exploit context histories in terms of context recognition. Particular contributions of the architecture are to enable a continuous, unsupervised learning of user behavior with life-long adaptation to changing environments and the use of nominal and ordinal sensor values in addition to numerical ones, effectively solving the issue of heterogeneity.

Context prediction is only one possible use of context histories, even if it might be the most challenging one. Nonetheless, there are many other uses that can provide additional benefit to the user of a system and that have already been analyzed in more depth by current research. Many of the issues discussed in this paper are also valid for other uses of context histories, and might thus be of help to research on those applications as well.

OPEN CHALLENGES

Most of the discussed issues are addressed by the presented approach to context prediction, but few are solved completely. Open challenges remain especially in:

- *Improving the accuracy of predicted contexts*: The time series prediction methods considered so far address sequential pattern prediction, but lack a detection of arbitrary periodical patterns and long term trends. Current developments like the algorithm presented in [2] to detect periodicities should be examined w.r.t. context prediction.
- *Coping with limited resources*: By applying online methods to context recognition and prediction, required storage and processing resources are generally low. However, eviction policies that are necessary to deal with strictly limited memory or real-time issues have not yet been considered.
- *Sharing context histories between devices*: As also mentioned in [3], sharing of context information can improve the accuracy of context recognition, and subsequently context prediction, by enhancing the view of the environment of each device with information that is not available locally. This is not necessarily limited to sharing only current context information, but could be extended to sharing complete context histories and predicted contexts.
- *Unobtrusive user interfaces for labeling context identifiers*: In our current work, a mapping of automatically recognized context classes, i.e. automatically constructed higher-level context identifiers, to descriptive context labels assigned by the user is assumed to be handled by the application. It is an open issue for HCI to design appropriate user interfaces for assisting this interactive process in an unobtrusive manner.
- *User acceptance*: No empirical user studies w.r.t. user acceptance of continuous context prediction have been conducted so far, but will be necessary before context prediction systems can be put into service for end users.
- *Uncertainty*: Dealing with uncertain sensor information on the one hand and with uncertain predictions on the other hand is currently not addressed satisfactorily. It is still an open issue for most time series prediction methods to compute measures of certainty alongside the actual predicted values.

Context prediction is a young topic, still at the outset of methodical research. When applied in a way that still leaves users in the loop of control, it can be a powerful tool to support users in their daily lives and to foster a broad availability of computing services to a larger public. However, the social implications of pervasive computing, and more specifically of exploiting context histories, must not be neglected; not only technological, but more importantly non-technological issues like a felt loss of control will rather sooner than later become urgent concerns.

ACKNOWLEDGMENTS

The author thanks Alois Ferscha for his initial ideas and several discussions about context prediction which helped to shape and clarify the concepts and the resulting software architecture. These concepts, the resulting software architecture and its implementation as a software framework have been developed in close cooperation with Harald Radi.

REFERENCES

1. Barkhuus, L., and Dey, A. Is Context-Aware Computing Taking Control away from the User? Three Levels of Interactivity Examined. In *Proceedings of the 5th International Conference on Ubiquitous Computing (UbiComp'03)*, 149—156, October 2003
2. Elfeky, M.G., Aref, W.G., and Elmagarmid, A.K. Using Convolution to Mine Obscure Periodic Patterns in One Pass. In *Proceedings of the 9th International Conference on Extending Database Technology (EDBT)*, 605—620, March 2004
3. Mäntyjärvi, J., Himberg, J., and Huuskonen, P. Collaborative Context Recognition for Handheld Devices, In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, 161—168, March 2003
4. Mayrhofer, R. An Architecture for Context Prediction. *Trauner Verlag*, Schriften der Johannes-Kepler-Universität Linz, volume C45, April 2005. (first appeared as *PhD Thesis*, Johannes Kepler University Linz, October 2004)
5. Penrose, R. The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics. *Oxford University Press*, 1989

Author Biographies

Rene Mayrhofer received his Dipl.-Ing. degree in Computer Science in 2002 and his PhD in Technical Sciences in 2004, both from the Johannes Kepler Universität Linz, Austria. He is currently working at the institute for Pervasive Computing as an assistant. His research interests include context awareness, embedded systems, peer-to-peer networks, artificial intelligence with specialization on spiking neural networks and security. He has served on the committees of PERVASIVE 2004 and PERVASIVE 2005 and is a contributing developer of the Debian GNU/Linux project.