QoC: Quality of Context – Improving the Performance of Context-Aware Applications

Tobias Zimmer

Telecooperation Office (TecO), Universität Karlsruhe
zimmer@teco.edu

Abstract. The research on context processing in ubiquitous computing environments has left the early state. So the next step on our journey towards the vision of Ubiquitous Computing will be to start building larger scale environments that contain applications and services from multiple sources and different providers. The goal of this work is to engineer a context management system that is able to help improving the context recognition rates of applications in large scale Ubicomp settings.

1 Introduction and Description of Purpose

The research on context processing in ubiquitous computing environments has left the early state. Today we have access to a selection of well understood context recognition techniques and algorithms, sensor nodes are available to implement and test context-aware application and there are specially designed communication systems on the marked that deliberately support the communication of context information in ad hoc networks. So the next step on our journey towards the vision of Ubiquitous Computing [1] will be to start building larger scale environments that contain applications and services from multiple sources and different providers. This step in our progress will yield a new class of problems in context handling that is induced by the scaling of the environments.

Today most context-aware systems have a comparably flat architecture. Sensor values are processed to contexts and this contexts are used to adept the behavior of appliances. Only very rarely we encounter settings that feature multi stage context processing, where context information is communicated and fusioned with other contexts to derive new ones. But this will most probably be the future of pervasive computing.

With my research I want to attack one of the problems caused by this advancement; namely high error rates in context recognition that can be ascribed to unmanaged multi stage context aggregation processes in large scale ubiquitous computing environments. Where scaling effects are subject to research in networking and sensor networks focusing on hardware and protocol issues, current research in ubiquitous and pervasive computing does not investigate scaling effects on context processing. Published context processing architectures (context models) are not capable of handling degression in context quality that is caused by a multi-step aggregation processes – most provide no tools for handling *context reliability* at all. Also we encounter a new class problems based on the fact that in highly dynamic modular environments we will not be able anymore to

Tom Pfeifer et al. (Eds.): Advances in Pervasive Computing 2006
Adjunct Proceedings of Pervasive 2006, Dublin, 7-10 May
books@OCG.at Vol. 207, ISBN 3-85403-207-2

determine "processing trees" – graphs that reflect the path and evolution of context information in an environment. One of these new problems is *cyclic usage of context*, another is *splitting and multiplication of context*. We will also have to cope with problem that have already been identified to be of relevance in advanced context processing like *locality of context* and *ageing of context* [2, 3].

2 Context-Management and -Attributes

The goal of this work is to engineer a context management system that is able to help improving the context recognition rates of applications in large scale Ubicomp settings.

The first step towards this context management system is to formalize the relevant architectural parameters of pervasive computing environments. This includes context, artefacts, communication schemas, processing schemas and quality parameters.

Quality of Context (QoC) is at the center of this work, it is defined by context attributes [4]. Context attributes are generic attributes of context data that apply to all contexts in general. That is context attributes are independent of the semantic interpretation of the context data, but still the attributes build a minimal ontology that allows common interpretation of the context in one important aspect: *Quality*.

Independence of the semantic interpretation is an advantage in system design. The separation of QoC handling and context processing enables the developer to choose the context model independent form the QoC system.

The focus in the selection of relevant context attributes has been twofold: Firstly the attributes have to provide an understanding of the QoC and they have to be generic, and secondly the number of attributes has to be as small as possible to guarantee that every artefact in an environment is able to implement the context management system in addition to its internal context processing logic. These requirements are matched by selecting the following four attributes: *spatial origin*, *age*, *reliability* and *degree of relationship*.

Spatial origin and age are attributes whose use to improve context recognition has been discuss in the community for some time. Likewise the use of a reliability measure is mentioned regularly. Using a measure of the degree relationship is new in this work. It addresses special issues in large ubiquitous computing environments that have been identified in the preparatory work for this thesis.

The spatial origin of a context is encoded using the RAUM system [5]. Age is encoded using two time stamps: one is indicating the time at which the oldest seed information in the context was sensed, the other indicates the time the current context was derived at. This approach allows to determine the time it took to derive a context from first sensing to its current state. Information on the spatial origin and age of a context allows the consumer to assess its relevance in advance to processing it [2].

Assessing reliability of context information yields one major problem: a statistically based measure like the recognition rate of a given context is of little or no help for the consumer because it only transports general performance information on the producer but no relevant information on the actual piece of context data it is associated with. This is a problem in context processing as the consumer needs to determine the individual quality of ever single piece of context to decide whether to process a context or not.

For most context recognition algorithms it is very expansive – in terms of computing power and energy consumption – to determine performance using methods from theory of errors. It might even be impossible if black box algorithms are used. In this case only statistical error rates can be determined in a classic mathematical model, again leading to the problem that no information on the current context is available.

In this thesis a reliability measure is established using a fuzzy logic expert system. The developer of a context-aware application typically has some experience on how her application performs under different input situations. In case of applications that aggregate context form sensor date, the data sheets of the sensory equipment are an additional source of information on reliability. This performance under different input conditions is split up into factors that can be described as fuzzy rules and associated with output functions building a Sugeno type fuzzy system. After inference of the fuzzy sets the crisp output value is communicated as a context attribute. On the consumer side this reliability measure is mapped to a lingual interpretation again and can be fed to the consumer's fuzzy systems.

The analysis of large application scenarios [6] and simulations of large scale Ubicomp setting has shown a class of problems connected to dependencies of data. Two of the major problems that were identified here are "*splitting and multiplication*" and "*cyclic usage*" as stated before. *Splitting and multiplication* can happen whenever a context is consumed by many producers of a similar kind. In this case a context is used in parallel by many appliances that all produce contexts of the same type.The newly produced contexts are then all derived from the same seed. If now an application seeks to consume this multiple available context type, it may be necessary to know whether theses pieces of context information are independent or not – e.g. if data fusion algorithms like Kalman filters should be used on the input contexts.

Cyclic usage means that context data is processed in a way that an artefact is consumer of a context that is derived form one it produced itself. This may not be a problem if the number of intermediate steps is high or the cycle is established intentionally, but in general it is not possible to rule out that cyclic usage of contexts is a problem for the applications. That is especially true if cycles from that conserve contexts over a long time without comprising new contextual information.

In the research work presented here, these typical problems of large scale ubiquitous computing environments are attacked by a completely new method, establishing a measure of the degree of relationship of contexts. This method – Genetic Relation of Context (GRC) – is based on ideas derived form biological genetics and genetic algorithms. When initially derived form sensor data, each context C is associated with a randomly generated bit-vector representing its genome. The genome Γ is then sequenced into genes γ.

$$\Gamma_C := (\gamma_{C,1}, \gamma_{C,2}, \cdots, \gamma_{C,n}) \tag{1}$$

Every time contexts are consumed and processed to derive a new output context, the genetic code of the source contexts is recombined to form the genome of the newly created context. To preserve the information of descent a specially designed probabilistic crossover method is used. The problem is that in contrast to biological genomes the context-genome has to be very small; e.g. *escherichia coli* an intestinal bacterium has about 4.500 genes encoded in $4,6*10^6$ base pairs each carrying 4 bit of information. This sums up to 2,19MB of information. The bit-vector representing the genome of a context typically carries 50-200bytes. This reduction is possible because, firstly the context-genome does not have to encode any functional information other than the descent where the biological genome stores lots more information, secondly the context genome can be designed to match the needs of context-aware environments in terms of the number of generations that have to be distinguishable and thirdly the crossover method can be adapted to emphasis the aspect of genetic relation. In genetic algorithms

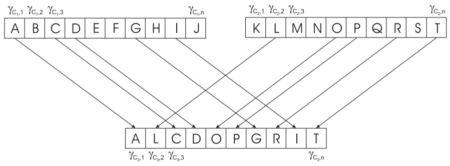

Fig. 1. Probabilistic multi site crossover (PMSC)

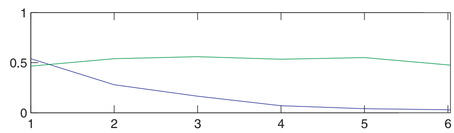

Fig. 2. Inheritance over six generations

mainly simple single site crossover methods are used that preserve schemata [7]. GRC uses a probabilistic multi site crossover method that preserves alleles of genes in loci. Probabilistic multi site crossover (PMSC), for each locus in a child's genome randomly chooses one of the genes occupying the same locus in the parents' genomes and inserts it (figure 1). This method allows very fast and computational inexpensive determination of the degree of relationship of contexts:

$$f_i(\gamma_{C_1,i}, \gamma_{C_2,i}) = \begin{cases} 0 & : \gamma_{C_1,i} \neq \gamma_{C_2,i} \\ 1 & : \gamma_{C_1,i} = \gamma_{C_1,i} \end{cases} \tag{2}$$

$$rel(C_1, C_2) = \frac{1}{N} \sum_{n=1}^{N} f_n(\gamma_{C_1,n}, \gamma_{C_2,n}) \tag{3}$$

The resulting ratio can directly represent the ratio of information derived from the parent context. Figure 2 shows the values of $rel(C_1, C_i)$ (the lower curve) under uniform crossing in one line on inheritance. The upper curve represents the $rel(C_{i-1}, C_i)$ values; the relation of every offspring to its direct parent. PMSC works not only for inheritance from two parents but also more if necessary. The maximum number of parents is limited by the number of genes in the genome. Additionally it is possible to adapt the probability of selecting a gene form a parent – in standard mode PMSC uses a uniform distribution – to reflect the ratio of information a parent-context contributes to the newly derived child-context.

2.1 Contribution

The introduced context management system allows QoC to be evaluated and propagated in ubiquitous computing environments. By means of context attributes, applications can evaluate whether a certain piece of context information is processed or discarded. Processing only high quality contexts leads to a reduction of misclassifications and thus to higher quality of output contexts. Introducing a separate layer containing context management functionality that is located below the semantic processing of context can save energy. Artefacts can determine whether a context has to be processed by costly recognition and fusion algorithms and so can go to sleep mode if no processing is necessary. Keeping the number of context attributes small makes the system realizable on almost all artefacts. The system is solves special problems in large heterogeneous Ubicomp environments that lead to a decrease of context quality. Without a context management system solving these issues, dynamic unsupervised large scale setting of pervasive computing technology will most probably not be feasible.

3 Approach

The work on this thesis splits into different stages; figure 3 shows the organization. Starting point of this research work was the observation of decreasing context quality while building larger ubiquitous computing environments. This motivated engineering a system to manage the Quality of Context (QoC) and led to the central statement of this work:

The Quality of Context (QoC) in large scale ubiquitous computing environments can be enhanced by exploiting context attributes as a common measure of context quality and establishing a management system using this information to adept context processing.

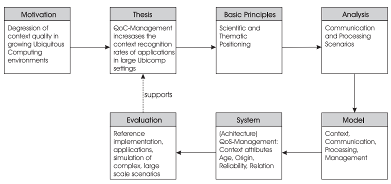

Fig. 3. Organisation of this thesis

The analysis part of this work is based on scenario studies [6] that where undertaken to get an idea of how future pervasive computing settings will look like and what requirements application in those environments will have to match. User experience form the applications running at TecO also add to the analysis that represents the base for formally modeling the aspects of pervasive computing environments relevant to this work. The model formalizes context, its processing, its communication and its management as well as the participants in an environment and their interaction. The architecture comprises the concrete realization of the system. It is basis for the implementation of the context management system as an extension to ConCom [8] running on Particle Computers [9]. Additionally this research will contribute a simulator for large scale ubiquitous computing environments that is used for developing the needed algorithms before implementing them on the actual hardware platform. The simulation also serves as evaluation environment for setting that can not be implemented in the real world due to high hardware costs and the lack of resources to manage a number of such large user studies as part of this work.

3.1 Evaluation

The evaluation of this work splits up in two parts. The first part is the evaluation in the real world. The context management system is implemented on the Particle Computer platform and gets integrated into the context-aware applications running at TecO. A survey will then compare context recognition performance of the applications running with the context management system enabled with the performance of the same applications running without the system. If time permits, I seek to additionally conduct small user study comparing the user satisfaction with these applications. The second part is the simulative evaluation of large scale settings. Again a survey will compare error rates in context recognition of applications running the context management system with those of applications not running the QoC system.

References

1. M. Weiser. The computer for the 21st century. *Scientific American*, 265 (3):66–75, 1991.
2. Albrecht Schmidt. *Ubiquitous Computing – Computing in Context*. PhD thesis, Lancaster University, November 2002.
3. Saul Greenberg. Context as a Dynamic Construct. *Human-Computer Interaction*, 16 (2-4):257–268, 2001.
4. Tobias Zimmer. Towards a Better Understanding of Context Attributes. In *Proceedings of PerCom 2004*, pages 23–28, Orlando, USA, March 2004.
5. Michael Beigl, Tobias Zimmer, and Christian Decker. A location model for communicating and processing of context. *Personal and Ubiquitous Computing Journal*, 6(5-6):341–357, 2002.
6. Vlad Coroama, Jörg Hähner, Matthias Handy, Patricia Rudolph-Kuhn, Carsten Magerkurth, Jürgen Müller, Moritz Strasser, and Tobias Zimmer. Leben in einer smarten umgebung - ubiquitous computing: Szenarien und auswirkungen. *Gottlieb Daimler- und Karl Benz-Stiftung*, 12 2003.
7. John H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.
8. Albert Krohn, Michael Beigl, Christian Decker, and Tobias Zimmer. ConCom - A language and Protocol for Communication of Context. ISSN 1432-7864 2004/19, University of Karlsruhe, 2004.
9. The Particle Computer Company. Website, accessed: 01/2006. `http://www.particle-computer.de`.