Using a Context Quality Measure for Improving Smart Appliances

Martin Berchtold
Christian Decker Till Riedel
TecO, University of Karlsruhe
Vincenz-Priessnitz-Str. 3, 76131 Karlsruhe
Germany

Tobias Zimmer
Particle Computer
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany

Michael Beigl
IBR
Muehlenpfordtstr. 23
38106 Braunschweig
Germany

Abstract

Many Ubicomp appliances require the recognition of context. Existing context systems do not provide information about the quality of the context recognizied to the appliance at runtime. In this paper we propose the first context quality system which gives quantitave measures, the Context Quality Measure (CQM), in real time. The CQM can be used by application to improve decision quality when interpreting context values. Our Fuzzy Inference System based approach considers the context detection algorithm as a black-box. It is therefore able to give generalized independet context quality measures and is applicable as an add-on to any context recognition system. A first practical implementation shows a gain of 33% in context detection quality in tested application scenarios.

1. Introduction

Many Ubiqutious Computing environments are built upon a collaborating set of smart appliances. Smart appliances consist of small computing devices integrated into everyday objects that provide some additional useful functionality either to the object itself and/or for the environment. Thus, smart appliances are required to detect the context of their use or the situation in the environment.

Some work has been carried out in developing context detection algorithms already (e.g. [6], [7] and [8]). Typically, context detection underlies a very varying quality: While in one situation the quality of the detection might be very high in other situations the same algorithm might deliver very bad results. A simple example from the AwareOffice Ubicomp setting should motivate the problem. The AwareOffice environment is a living laboratory office space that is used since several year to collect experiences and to perform studies in Ubiquitous Computing. One of the appliances within the AwareOffice is the a context aware whiteboard pen. This AwarePen appliance uses a pre-trained context recognition

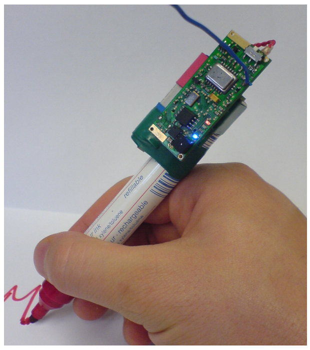

Figure 1. Writing with the AwarePen.

system to detect situations including "writing", "playing" and "laying still". An accelerometer integrated in a sensor node attached to the pen recognises raw movement which are then interpreted by software on the node (see Fig. 1). The detected situation information is then distributed to other appliances in the AwareOffice environment. We experienced that certain movement patterns are very simple to classify, e.g. separating writing or playing situations. Other movement patterns - e.g. produced by other users having a different style of using the pen while writing - are much more difficult to classify. Another important experience was that changes in the quality of detection could happen in a very short time span: A user writing a text on the board, then for some seconds playing with the pen when thinking and then continuing writing is a typical example for such a situation. For a context recognition algorithm, such changes in the movement pattern are difficult to classify and therefore lead to a low detection quality for the context. We also found that it is essential for a Ubicomp application to have knowledge about the quality of the detection. In our example, the context received from the pen is used by the camera of the whiteboard to take a picture copy of the content when a writing session was over. The action of taking a picture was triggered on the context detected by other appliances in the environment, including the AwarePen. Thus, to allow for a high quality of the whiteboard camera decision, a quality measure for the context input is required.

Today, to our knowledge, all existing context recognition systems are not able to deliver an independent quality level for detected context in real-time. This paper introduces the first context algorithm and system that is able to provide a real-time quality level for a context and situation recognition step. We introduce a Context Quality Measure (CQM) as a continous measure as a numerical value between 0 (wrong) and 1 (correct). Approach and measure are independent of the used context recognition method thus ensuring general applicability as an add-on for any context recognition system. We consider the context algorithm as a black-box where our context quality system could be added to. The final context recognition system then consists of the existing context algorithm and our context quality system (2).

2. Architecture

Online context classification assigns a set of sensor cues to a context class or context c (2). Each cue represents a single sensor. Cues are computed from sensor data and identify basic features for the context classification. The quality of an algorithm is a state dependent analysis of the relation of the input cues and the classified context. The input of the quality analysis uses the same sensor cues as the context classifier, plus context c. The result of the quality analysis is the CQM q. The context quality system considers the recognition algorithm as a black box. This way the design is applicable to all recognition algorithms.

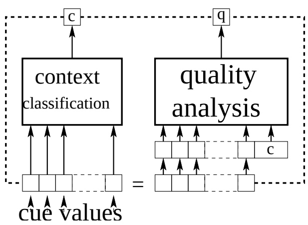

Figure 2. General Interconnection of contextual classification and qualitative measure.

2.1. Context Quality System

We consider the quality measure as totally independent from the system that carries out context recognition. The value of the quality measure is based on the current contextual state and the sensor date only. This is the same state the contextual state c was determined on. Prior knowledge whether a contextual classification was right or wrong is stored in a fuzzy inference system (FIS).

2.1.1 Interconnection of Context Recognition and Quality Measure

The input for analyzing the reliability is a vector that is the combination of cued sensor values and identification of the contextual state. The n-dimensional input of the universes $\mathcal{V}_1, .., \mathcal{V}_n$ for a contextual classification is defined as follows:

$$\vec{\mathbf{v}}_C := (v_1, .., v_n), \text{ with } n > 1$$

With an identification c of an arbitrary contextual class the input vector for the system that does the quality measure is defined in a following way:

$$\vec{\mathbf{v}}_Q := (\vec{\mathbf{v}}_C, c) = (v_1, .., v_n, c)$$

Each time the contextual classification gets a new input $\vec{\mathbf{v}}_C$, the classification result is combined with this vector in a new vector $\vec{\mathbf{v}}_Q$. The reliability estimation is based on the vector $\vec{\mathbf{v}}_Q$, which is insofar the interconnection between the contextual and the qualitative system. The determination of the quality measure is done by a fuzzy inference system (FIS) that holds the information about the correctness of the contextual classification on the base of the input $\vec{\mathbf{v}}_C$.

2.1.2 TSK Fuzzy Inference System (FIS)

Takagi, Sugeno and Kang [10][9] (TSK) fuzzy inference systems (FIS) are fuzzy rule-based structures, which are especially suited for automated construction. With the TSK-FIS the consequence of the implication is not a functional membership to a fuzzy set but a constant or a linear function. In our system the linear functional consequence is used, since the results for the reliability determination are better. The linear functional consequence of the rule j depends on the input of the FIS:

$$f_j(\vec{\mathbf{v}}_Q) := a_{1j}v_1 + a_{2j}v_2 + \ldots + a_{nj}v_n + a_{(n+1)j}c + a_{(n+2)j}$$

The membership functions are non-linear Gaussian functions, which are calculated accordingly:

$$F_{ij}(v_i) := e^{-\frac{(v_i - \mu_{ij})^2}{(2\sigma_{ij}^2)}}$$

The rule j of the TSK-FIS is verbalized in a linguistic form as follows:

$$\text{IF } F_{1j}(v_1) \text{ AND .. AND } F_{(n+1)j}(c) \text{ THEN } f_j(\vec{\mathbf{v}}_Q)$$

The antecedent part of the rule j determines the weight w_j accordingly:

$$w_j(\vec{\mathbf{v}}_Q) := \left(\prod_{i=1}^{n} F_{ij}(v_i) \right) F_{(n+1)j}(c)$$

The projection from input $\vec{\mathbf{v}}_Q$ onto the quality measure $\widehat{q}$ is a weighted sum average, which is a combination of fuzzy reasoning and defuzzification. The weighted sum average is calculated according to the rules $j = 1, .., m$ as follows:

$$\mathbf{S}_{\widehat{Q}}(\overrightarrow{\mathbf{v}}_Q) := \frac{\sum_{j=1}^{m} w_j(\overrightarrow{\mathbf{v}}_Q) f_j(\overrightarrow{\mathbf{v}}_Q)}{\sum_{j=1}^{m} w_j(\overrightarrow{\mathbf{v}}_Q)}$$

The TSK-FIS $\mathbf{S}_{\widehat{Q}}$ maps onto a set $\widehat{Q}$, which is not a desirable quality measure since its boundaries can not be determined. The value $\mathbf{S}_{\widehat{Q}}(\overrightarrow{\mathbf{v}}_Q)$ needs to be normalized to fit in a designated set Q of quality measures.

2.1.3 Normalization of FIS Result

The mapping of the TSK-FIS is not restricted to a certain interval due to the automated construction process. The designated output for the TSK-FIS is zero for a false and one for a right contextual classification. The error that cannot be fully eliminated in the training process, represents the error of the contextual classification in a relative manner. The distance between the designated $\{0, 1\}$ and the actual output $\mathbf{S}_{\widehat{Q}}(\overrightarrow{\mathbf{v}}_Q)$ is the representation of the contextual classification error. This means that the quality measure is not only an indicator if the contextual classification was right or wrong, it also shows how right or wrong the classification was. The error between designated and actual output is distributed around one and zero, so values over one and under zero are possible. These values need to be normalized to the interval $Q = [0, 1]$ of the desired quality measure. The normalization is done via a function L that maps onto the interval $Q = [0, 1]$ or an error state ϵ. An error state ϵ represents the quality measures, which can not be mapped onto the interval $Q = [0, 1]$ in a semantically correct way. The values lower than $-0, 5$ would represent an error for the designated output one after the normalization. A semantically interpretation of the value is, that it belongs to zero with an error of mapping. These circumstances are the same for values over $1, 5$ and the designated output one. So values under $-0, 5$ and over $1, 5$ are mapped with the function L onto the error state ϵ. With these facts the normalizing function L is defined as follows:

$$L(x) := \begin{cases} x & \text{if } 0 \leq x \leq 1 \\ -x & \text{if } -0,5 \leq x < 0 \\ 1 - x & \text{if } 1 < x \leq 1,5 \\ \epsilon & \text{else.} \end{cases}$$

The quality measure is computed by the TSK-FIS $\mathbf{S}_{\widehat{Q}}$ concatenated with the normalizing function L in mapping $\mathbf{S}_Q$ onto the set Q:

$$\mathbf{S}_Q : \begin{cases} \mathcal{V}_1 \times ... \times \mathcal{V}_M \times C & \longrightarrow \quad Q \cup \epsilon, \text{with } Q = [0, 1] \\ (v_1, .., v_n, c) & \longmapsto \quad L \circ \mathbf{S}_{\widehat{Q}}(v_1, v_2, .., v_m, c) \end{cases}$$

The system to calculate the (normalized) quality measure is referred with the function $\mathbf{S}_Q$ from now on. This then results in the final measure of the CQM q.

2.2. Automated Construction of FIS

The structure of the TSK-FIS $\mathbf{S}_{\widehat{Q}}$ is determined by an automated method that consists of a fuzzy clustering, a linear regression analysis and the training of a neural fuzzy network. The designated output - that is needed to converge with the atomization process to - is one for a right and zero for a wrong contextual classification. The data the automated creation process needs to adapt, is a set of input vectors that were contextually classified. The designated output is assigned to each of the samples.

2.2.1 Structure Identification with Fuzzy Clustering

There are several algorithms of fuzzy clustering. Since there is no knowledge about how many clusters there are, a algorithm is needed that determines the number automatically. A mountain clustering [11] could be suitable, but is highly dependent on the grid structure. We opt for a subtractive clustering [2] instead. This clustering estimates every data point as possible cluster center, so the prior specifications are none. A definition of parameters the subtractive clustering needs for good cluster determination are given by Chiu [3]. The subtractive clustering is used to determine the number m of rules, the antecedent weights w_j and the shape of the initial membership functions F_{ij}. Based on the initial membership functions a linear regression can provide the consequent functions

2.2.2 Linear Regression with Least Squares

The weights a_{ij} of the consequent functions f_i are calculated through a linear regression. The least squares method fits the functions f_i into the data set that needs to be adapted. A linear equation for the differentiated error between designated and actual output - which can be calculated with the rules and initial membership functions the subtractive clustering identified - is solved for the whole data set with a numeric method. The single value decomposition (SVD) is used to solve the over-determined linear equation. With the initial membership functions F_{ij}, the rules j and the linear consequences f_j a neural fuzzy network can be constructed. The neural fuzzy network is used to tune the parameters a_{ij}, μ_{ij} and σ_{ij}^2 in an iterative training towards a minimum error.

2.2.3 Adaptive-Network-based FIS (ANFIS)

A functional identical representation of a FIS as neural network is an Adaptive-Network-based FIS (ANFIS) [5]. The structure of a ANFIS for the qualitative TSK-FIS $\mathbf{S}_{\widehat{Q}}$ is visualized in figure 3. Most of the networks neurons are operators (circled symbols) and only the squared functions are adaptable neurons. This neural fuzzy network is used to tune the adaptable parameters a_{ij} of the linear consequents and

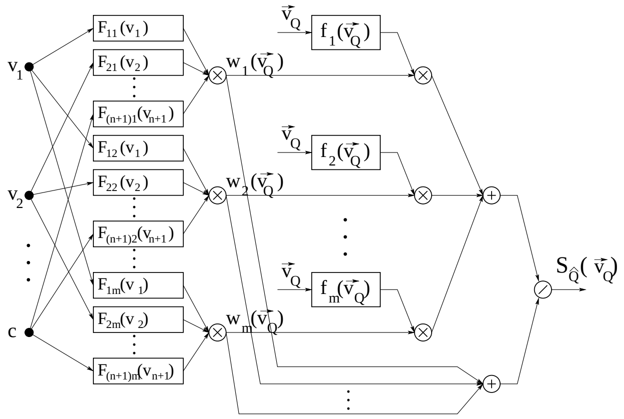

Figure 3. ANFIS for qualitative TSK-FIS.

μ_{ij} and σ_{ij}^2 of the gaussian membership functions. The tuning process is done iteratively trough a hybrid learning algorithm.

2.2.4 Hybrid Learning

The hybridity of the learning algorithm consists of a forward and a backward pass. In the backward pass a backpropagation of the error between designated and real output of the ANFIS to the layer of the Gaussian membership functions is carried out. The backpropagation uses a gradient descent method, that searches a preferably global minima for the error in an error hyperplane. The forward pass performs another iteration of the least squares method with the newly adapted membership functions of the backward pass. The hybrid learning stops for the data set used when a degradation of the error for a different check data set is continuously observed. The resulting ANFIS represents the qualitative non-normalized TSK-FIS $\mathbf{S}_{\widehat{Q}}$.

2.3. Statistical Analysis

Statistical analysis is used to determine how the probabilistic odds are to separate the correct from the wrong classifications through the quality measure.

2.3.1 Maximum Likelihood Estimation (MLE)

With a maximum likelihood method the normal distributions of the quality measure for right and wrong classified data points are estimated. A small data set for testing the behavior of the quality measure is not significant enough to calculate a statistical mean or a standard deviation. Therefore, an estimation of the mean values and the standard deviations is more statistically confirmed. With a infinite test data set the mean value and the standard deviation of the Gaussian density function are the same for both methods. The maximum likelihood estimation (MLE) requires knowledge for each data point, if its classification was correct or wrong. This requires a second data set different from the training set, which also contains information on the correctness of the classification. This second data set is now used for the statistical analysis in order to derive the Gaussian density functions. The functions for right (r) or wrong (w) classifications are defined as follows:

$$\varphi_{\mu_{r|w},\sigma_{r|w}}(x) := \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu_{r|w})^2}{2\sigma_{r|w}^2}}$$

2.3.2 Threshold Determination

The threshold value s for the quality measure, that is used to separate the data, can also be determined via a MLE for a data set without secondary knowledge. The mean value of the estimation is than the definition of the threshold s. Since the estimation was done for a data set with secondary knowledge of correctness of the contextual classification another estimation is not required. To visualize and validate the quality measure of the test data set this secondary information is needed anyway. The threshold s is now determined through the intersection of the two Gaussian density functions. For a infinite data set the MLE without secondary knowledge and the intersection method converges.

2.3.3 Probabilities

With the density functions and the median cuts through the threshold s the probabilities for right and wrong classifications can be calculated. The median cut from negative infinity to the threshold value for the Gaussian density functions can be calculated as follows:

$$\Phi_{\mu_{r|w},\sigma_{r|w}}(\overline{s}) := \int_{-\infty}^{s} \varphi_{\mu_{r|w},\sigma_{r|w}}(x)dx$$

The median cut for the other half of the density function is calculated accordingly:

$$\Phi_{\mu_{r|w},\sigma_{r|w}}(\underline{s}) := \int_{s}^{\infty} \varphi_{\mu_{r|w},\sigma_{r|w}}(x)dx$$

With these median cuts the probability $\mathbb{P}$ of selecting right classifications c through a quality measure q over the threshold s is calculated accordingly:

$$\mathbb{P}(c = right|q > s) = \Phi_{\mu_r,\sigma_r}(\underline{s}) - \Phi_{\mu_w,\sigma_w}(\underline{s})$$

The probability $\mathbb{P}$ to select true negative classification, i.e. $c = wrong$ is:

$$\mathbb{P}(c = wrong|q < s) = \Phi_{\mu_w,\sigma_w}(\overline{s}) - \Phi_{\mu_r,\sigma_r}(\overline{s})$$

The probabilities for the false negative and the false positive quality measure indications are:

$$\begin{aligned} \mathbb{P}(c = right|q < s) &= \Phi_{\mu_r,\sigma_r}(\overline{s}) \quad \text{and} \\ \mathbb{P}(c = wrong|q > s) &= \Phi_{\mu_w,\sigma_w}(\underline{s}) \end{aligned}$$

3. Implementation

For evaluating the quality measure a physical device was needed. The applications of the AwareOffice [13] offer a platform to acquire physical data and a variety of contextual classification algorithms in use. The AwarePen appliance was chosen since the usability was conclusive.

3.1. AwarePen

The AwarePen is a whiteboard marker that can detect the contextual states ”lying still”, ”writing” and ”playing around”. For contextual classification a TSK-FIS is used that maps standard deviations from three acceleration (aka adxl) sensor outputs onto context classes. Indicator of the class and three standard deviations are used in the quality analysis to determine the reliability of the classification dynamically. All parts of the processing cue of the AwarePen are shown in figure 4.

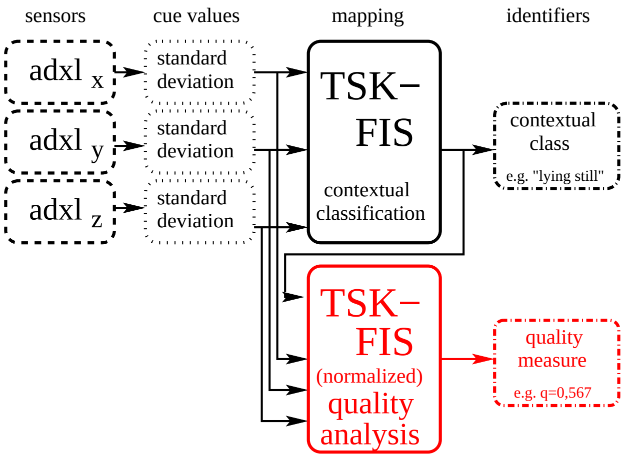

Figure 4. Schematics of AwarePen.

3.2. Evaluation

For evaluation purpose a small test set of 24 data points was created. The quality measure for the contextual classifications are shown in figure 5 with their actual rightness in symbols (o right and + wrong classifications). The statistical mean values for right (grey) and wrong (black) classifications are plotted in dashed lines. The density functions are plotted in figure 6 where the median cut of the threshold values are marked with a green line and hatched areas. The optimal quality threshold of $s = 0,81$ is located at the intersection of both density function (marked with a green line) and was derived using the foundations from section 2.3.2. At this optimum the probability for the test set that the quality measure is above the threshold, meaning the classification was correct, is equal to the probability that a quality measure is below the threshold $\mathbb{P}(c = right|q > s) = \mathbb{P}(c = wrong|q < s) = 0,8112$. The probability that a quality measure indicates a correct classification that was actually wrong is $\mathbb{P}(c = wrong|q > s) = 0,0217$. The probability that a wrong classification that was actually right is $\mathbb{P}(c = right|q < s) = 0,0846$. In the test data set the correct classifications are fully separable from the wrong contextual classifications. Results indicate that the appliance can discard 33% of the classifications, which equals all wrong contextual classifications, when using the quality measure. The separation has not always to be that clear. For a large set of data the odds for separating the data are worse. The evaluation shows that a quality measure combined with a statistically estimated threshold can help separating good from bad contextual classifications. The threshold in the shown example is not in-between the highest (one) and the lowest (zero) quality measure but closer to the highest. This reflects the error of the context recognition, which was adapted through the automated construction method. If the training set has equal amount of right and wrong samples the quality measure would lead to a threshold $s \approx 0,5$.

4. Related Work

The term Quality of Context (QoC) was coined in numerous works by various authors such as [1]. We focus on one dimension of QoC related to the ”probability of correctness” [1]. In contrast to our approach related work often restricts itself to constant probabilistic measures for algorithmic errors or sensor failure. We are not aware of any previous work in ubiquitous computing that explicitly uses or generates quantifiable quality measures on the basis of a generic predefined algorithm. Fuzzy logic in general has been sparsely used in other context reasoning systems. In those systems the quality can be seen as implicitly modeled by the uncertainty. It can, however, be observed that system like [4] use fuzzy inference on higher levels of context processing. The concept of a reliability measure underlying this paper was first proposed in [12]. [12] and [1] give a more complete argumentation for the necessity of quality measures in context reasoning.

5. Conclusion and Outlook

We presented a context quality system generically applicable to context recognition algorithms. A quality measure

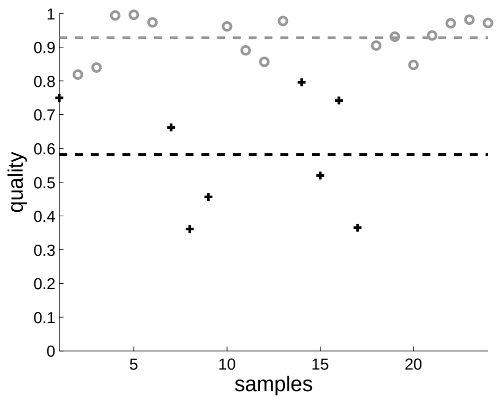

Figure 5. Quality measure for test set with 24 data points for right (o) and wrong (+) contextual classifications and statistical mean values (dashed lines).

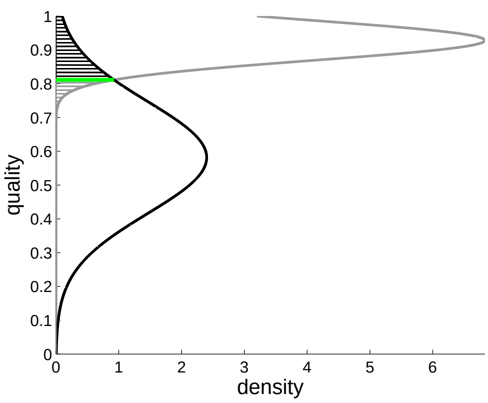

Figure 6. Gaussian density functions for right (grey) and wrong (black) classified data with marked threshold value (green) and hatched median cuts.

- the output of the quality system - provides real-time additional information on the context classification for smart appliances. The optimal quality threshold was derived for comparison with the quality measure in order to decide on the acceptance for a context classification. As a result and demonstrated in the AwarePen appliance, the quality system could be successfully applied to filter out low quality context therefore improving the decision quality of the application by 33% in our example. We implemented and tested the quality system on the AwarePen, a typical Ubiquitous Computing artefact consisting of a whiteboard pen that was augmented with a Particle Computer as sensing and computing platform. This rather high improvement of quality is backed up by other applications build in the AwareOffice. We are in the process of integrating the context quality system to other appliances and testing the system.

Future research will cover the use of the context quality system for context prediction. The quality measure can i.e. indicate that a context classification changes in direction to another context. Our research will also look into how to support support fusion and aggregation for higher level contexts that may be able to classify complex situations. Such complex context systems may unveil the true potential of Ubiquitous Computing and context aware systems in the future. In order to process reasonable output, higher level context processors requires a measure to decide which of the simpler context information to belive. Our general Context Quality Measure (CQM) concept is able to build the basis for such decisions. We will continue our research here studying practical applications and theory.

References

[1] T. Buchholz, A. Kuepper, and M. Schiffers. Quality of context: What it is and why we need it. *Workshop of the HP OpenView University Association*, 2003.

[2] S. Chiu. Method and software for extracting fuzzy classification rules by subtractive clustering. *IEEE Control Systems Magazine*, 1996.

[3] S. Chiu. *Fuzzy Information Engineering: A Guided Tour of Applications*, chapter 9, Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification. John Wiley&Sons, 1997.

[4] A. A. Eldin and R. Wagenaar. A fuzzy logic based approach to support users self-control of their private contextual data retrieval. *European Conference Information Systems*, 2004.

[5] J.-S. R. Jang. Anfis: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man and Cybernetics, 1993, vol. 23 pp. 665-685*, 1993.

[6] K. V. Laerhoven, K. Aidoo, and S. Lowette. Real-time analysis of data from many sensors with neural networks. *ISWC, pp 115-123*, 2001.

[7] R. Mayrhofer. Extending the growing neural gas classifier for context recognition. *WHPS at EUROCAST*, 2007.

[8] K. Partridge, J. Begole, and V. Bellotti. Evaluating contextual models. *UbiPCMM*, 2005.

[9] M. Sugeno and G. Kang. Structure identification of fuzzy model. *Fuzzy Sets Systems, 1988, vol 26(1), pp 15-33*, 1988.

[10] T. Tagaki and M. Sugeno. Fuzzy identification of systems and its application to modelling and control. *IEEE Trans. Syst. Man and Cybernetics, 1985, vol 15, pp 116-132*, 1985.

[11] R. Yager and D. Filer. Generation of fuzzy rules by mountain clustering. *Journal IFS, vol 2, pp 209-219*, 1994.

[12] T. Zimmer. Qoc: Quality of context - improving the performance of context-aware applications. *APC*, 2006.

[13] T. Zimmer and M. Beigl. Aware office: Integrating modular context-aware applications. *IWSAWC*, 2006.