Electronic Communications of the EASST
Volume X (2010)

Proceedings of the
Third International DisCoTec Workshop on
Context-Aware Adaptation Mechanisms for
Pervasive and Ubiquitous Services
(CAMPUS 2010)

Training the Behaviour Preferences on Context Changes

Kuderna-Iulian Bența, Marcel Cremene and Amalia Hoszu

12 pages

Guest Editors: Sonia Ben Mokhtar, Romain Rouvoy, Michael Wagner
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Training the Behaviour Preferences on Context Changes

Kuderna-Iulian Bența, Marcel Cremene and Amalia Hoszu

Technical University of Cluj-Napoca

Abstract: Personalized ambient intelligent systems should meet changes in user's needs, which evolve over time. We propose *BAM – * Behaviour Adaptation Mechanism, a neural-network based control system that is trained, supervised by user's (affective) feedback in real-time. The system deduces the preferred behaviour, based on the detection of affective state's valence (negative, neutral and positive) from facial features analysis. The neural network is retrained periodically with the updated training set, obtained from the interpretation of the user's reaction to the system's decisions. The *BAM mechanism is implemented in the Affective-aware Smart Home (ASH), a multi-agent and ontological context-based system. We implemented a simple example consisting of a control system to position the blinds according to the inside and outside light level. We investigated how many training examples, rendered from user's behaviour, are required in order to train the neural network so that it reaches an accuracy of at least 75%. We present the evolution of behaviour preference learning parameters when the number of context elements increases.

Keywords: Ambient Intelligence, Affective Computing, Personalization, Context Awareness, Ontology, Neural Networks.

1 Introduction

Intelligent ambient systems aim is to help the user to manage the various devices surrounding him. An intelligent ambient (IA) system like a smart home should have the ability to respond to individual needs [KMGA08]. Also, such a system should be non-intrusive [CEF09].

Objective. Our objective is to create an IA system that observes the user reactions and learns from these observations. A non-intrusive way to observe the user it is to monitor his facial expression. Our system deduces the preferred behaviour, based on the detection of affective state's valence (negative, neutral and positive) from facial features analysis.

Scenario. In order to have a better understanding of our problem we present an application scenario. The A part, concerns the learning of a new behaviour preference and the B part concerns the situation when a new context element is added.

A. Maria is an old and speech impaired person. She is invited to her friend Laura that has an Affective-aware Smart Home (ASH). As Maria is a welcomed guess, the system will authorize her to personalize the system's behaviour. One of the house behaviour rules closes the blinds when the outside light has the same intensity as inside. Maria likes to look outside the window and so, when the first decision of the system to close the blinds is triggered (at sunset, for instance), she will display immediately (in the following minute) a negative emotion (i.e.

anger), showing her disapproval. The ASH will learn (after repeating it a few times, if needed) the Maria's new preference.

B. Later on, ASH is upgraded with a temperature sensor that senses the room and outside temperature. By expressing her reaction to the system's decisions, Maria is effectively providing new behavioural preference examples and the upgraded ASH learns to react to both, light and temperature context elements.

Technical questions. For achieving our objective, we need to answer to the following questions:

1. How should we represent the ambient knowledge?
    1.1. How should we represent the context including user data?
    1.2. How should we represent the user preferences?
2. How are the user's behaviour preferences discovered?
    2.1. How do we get and interpret the user's response?
    2.2. How does the preferred behaviour learning mechanism work?
3. How should the system adapt its behaviour to the user's needs of preferences?

Approach. We propose *BAM, a control mechanism that allows the system to learn the new behaviour preferences without editing the rules by hand, but feeding back the user's multiple kind of responses (symbolised by the "*" preceding the acronym "BAM"), to the system's decisions.

In order to prove the concept we captured the user's affective reaction from facial displays with "FaceReader", the personalized version of it [BKE+09], to read at least three valence levels that work as positive, negative or neutral feedback. The results in training a MLP neural network to learn the preferred behaviour from the user's affective reaction are discussed. Context ontology is used to describe the user context and preferences.

Outline. The rest of the paper is organized as follows. In section 2 we overview the existing solutions regarding the aforementioned questions. In section 3 we present the principle of the user reaction controlled loop mechanism, *BAM, to online learn the behaviour preference. The implementation details of the Affective-aware Smart Home (ASH) with the ☺BAM variant are explained in section 4. In the next section we analyse the accuracy of the MLP neural network to learn the preferred behaviour from the user's affective reaction when the context changes. In the last section we conclude our work and present the future work.

2 Related Work

This section aim to respond to the technical questions described in the introduction. We have analysed a series of approached that involves learning.

User needs are evolving over time. In order to meet this requirement, one option is to let the user edit the behaviour rules in a GUI. But editing the IA rules is difficult for the user because of the complexity that comes with the use of different sensors and actuators, leading to a large number of rules to define. [NYS+05][GYC+07]. Moreover in [GYC+07] the authors notice that "rule-based reasoning is not flexible and can not adapt to changing circumstances". A second option to determine the user's preferences is to use machine learning techniques. A third, hybrid option, is presented in [MTWP09], where the authors propose a combination of rules and machine learning to personalize the behaviour of the system. Although this solution seems promising, it is yet unclear what would happen when the rule set will grow larger.

We decided for the second option, considering that impaired persons find it difficult to give some direct commands (vocal or physical) in order to control the behaviour of a smart home, but they may facially display short time responses like affective states (emotions), do gestures or actions that could be used as feedback. For instance, taking as reference the normal neutral valence affective state, a positive displayed emotion will mean approval and a negative emotion will mean disapproval of the system's decision, if expressed immediately after it.

Another less addressed issue is the adaptation to context structural changes. In [MPTW09] a rule based solution is presented to tackle the problem which appears when increasing the possible values that a context element can take. Also, the number of context elements may change due to upgrades in an IA system, challenging it to become scalable [ACPP+09].

Behaviour preference representation. In [HIR06], the authors present a review of the existing context related preference representation. Also, they propose a score based solution. They assign a score to each preference possibility, consisting in a real value in the [0, 1] interval or a predefined value (veto, indifferent, mandatory, error situation). If a context *C*, and an associated variable set *v* are present, the score will be the function *score(p.s,C,v)*, where *p.s* is the scoring expression, otherwise the score is indifferent. In this model the context elements are considered distinct, without any relation between them.

An ontological representation of the preferences is presented in [ALL05]. It models ontologically the relations between the context elements and the preferences. The Preferences class has relations with all the main classes (*Time*, *Agent*, *Location*, *Activity*). The preference can be positive or negative indicating an appropriate or inappropriate choice for a resource, environment or operation. This model uses a probability to set the preference priority, but has only two values to express the relation between the context and the service (desired behaviour).

Another solution [HAM+06] uses Bayes RN-Meta-Networks, organized in multi-layers. The preferences are modelled by complex levelled conditional probabilities between the user, the context and the preferred service.

In the article [Fla05] the author presents an associative network between context and application. Each context element could be associated with all N applications for a user. The association relation is modelled by a variable weight *w* that indicates the connexion strength between the context element and the application, thus given the weight matrix and a certain context, one may predict the application a user will choose. Extending this idea, the weights could store the user's preferences, but lacks the advantages of ontological modelling.

The neural networks are used in [SKW05] to describe weighted relations between the context elements (responding to: who, where, when, how) and the context elements (responding to how), the services and service parameters. They use MLPs (Multi Layer Perceptrons) with one hidden layer. This solution does not use ontologies in context modelling.

Solution	Ontological Context	Context-service relation
CtxPrefScore [HIR06]	-	score ([0, 1])
OWLPref [ALL05]	+	ontological (appropriate/not)
Bayes Meta-Nets [HAM+06]	-	probabilistic
NNAssoc [Fla05]	-	association network weights
UPM [SKW05]	-	MLP weights

Table 1: Comparison between different preference representation solutions.

We may notice in Table 1 that only one solution adopted an ontological context modelling and has only two values to express the relation between the context and the desired behaviour.

Behaviour adaptation mechanisms. There are different approaches for context-service (behaviour) relation which allow for a more or less fine grained expression of the preferences. The Bayes Meta Network solution [HAM+06] is the nearest to meet our online updatable preferences objective, in the sense that it uses user's feedback, but in this case they do not use ontologies and need a prior probabilities calculation.

Regarding the use of emotional response for learning the desired behaviour, the article [Bro07] presents a reinforcement learning mechanism where a social robot learns from rewards and punishments expressed by positive (happy) and negative (fear) emotions. A reinforcement learning mechanism implies giving feedback for a set of tasks, but our objective is to have a simpler loop with immediate response. We also searched for a more general emotion valence assessment tool, explained in detail in [BKE+09] where we proposed a personalized version of the FaceReader [FR] for detecting the user's affective state valence.

Solution	Machine Learning	Learning type	Feedback	On/off-line
Adaptive Home [Moz05]	Q-learning	reinforcement	implicit	online
Bayes Meta-Nets [HAM+06]	Bayes Meta-Nets	supervised	explicit	online
FLORA-MC [SHRS08]	FLORA-MC	supervised	implicit	online
CASAS [RaC09]	HMM	supervised	both (i&e)	offline

Table2: Comparison between different preference machine learning techniques.

Because the user's implicit reaction (from the historical data) can be intrusive, the explicit feedback is preferable [HAM+06] [RaC09]. Bayes Meta Network solution [HAM+06] supports online preference discovery mechanism in context awareness. The mechanism consists in updating the preference model for each user if the system's decision was disproved by at least one user. The preference model update is done by calculating the distribution probability for each user and then propagating the values to the next meta-network layers. Its main issue is that the prior probabilities need to be initially calculated by a human that is difficult for a large number of context elements. The main advantage of this model is that it supports online preference update.

There are preferences learning solutions that also allow online adapting when changes in user's preferred system behaviour occurs by relearning the preferences [Moz05] [HAM+06] [SHRS08]. CASAS [RaC09] handles this problem indirectly by observing the changes in activity patterns (activity start time, duration) making predictions about the action that the user will do in a house (e.g. to turn on the TV, the lights, etc.) in a certain temporal context that repeats in a similar way, periodically.

The learning parameters of the modified preferences (re-learning), like the number of necessary examples for training, the time necessary for applying the preferences (feed-forward), are rarely discussed (only [SHRS08]). Moreover, [MPTW09] addresses preference learning when possible values of one context element vary.

We describe a supervised preference learning mechanism based on explicit feedback and we analyse it in section 5, similar to [SHRS08], the ☺BAM learning parameters.

In [MPTW09], it is discussed the problem of preference update when possible values of a single context element increases. Further more, our approach is original in the sense that it tries to answer an intriguing question: what happens with the learned preferences, when the number of context elements changes.

After analysing these examples, our conclusion are:

1. Ontological representation of the user preference in context-aware systems is rarely addressed despite the advantages of using ontologies.
2. The behaviour adaptation mechanisms are online and supervised, rarely use explicit user feedback. Also, these solutions rarely handle preference when the context structure changes, like different number of values for a context element or variations in the number of context elements.

3 The Principle of * Behaviour Adaptation Mechanism

We made the following decisions for modelling the intelligent ambient (IA), in particular ASH's knowledge:

1. To use the ontology for context and service representation
2. To represent the context-service relation, that is the preferred behaviour as weights, stored in the ontology
3. To update the preferred behaviour according to the user's feedback to the system's decisions

In principle, we consider the context C, composed by context elements in relation with each other, a service vector S, and a weight vector w, that records the preferred behaviour, that is the service to choose when the context C is present and a current reaction $\mathcal{R}$ of the user U.

The meaning of the "*" preceding BAM is that this mechanism is acquired though multiple type of feedback, explicit: voice commands, affective states, GUI-based or implicit: analysing the human behaviour. The affective "☺" variant of *BAM is explained in section 3.2.

3.1 Preference representation

We argue that storing the preference in neural network weights is better then in Bayes RN Meta-networks like in [HAM+06] because:

1. The neural network allows initial training by an example training set, comparing to a mandatory prior probability calculations, simplifying the work at this stage.
2. If the rules or the Bayesian approach would be used, a full description of the behaviour should be given (all combinations of context values and desired behaviour), a neural network can run with a few training examples if any, due to its generalization capability, and adjust online.
3. The neural network has the ability to generalize from a given set of examples.

Representing the preferences in ontology is motivated by the following arguments:
1. The ontology supports the distribution and reuse of the once learned preference in other applications with the same context elements and services, or similar (when increasing or decreasing one or more context elements or services).
2. The neural network is to become dynamically reconfigurable (we may change its parameters on runtime: the number of hidden layers, neurons on each layer, activation function type, learning rate for each layer neurons).

The part of the ontology containing the representation of the neural network is beyond the purpose of this article, as in this first implementation we saved the neural network parameters values in a file.

3.2 The Affective Behaviour Adaptation Mechanism

We propose to replace the rule based decision mechanism with a neural network that learns from the user's feedback the new preferred behaviour in order to respond to the user's new needs. Because we like the user to interact as natural as possible with the system, we propose to use the affective kind of user feedback (☺BAM). To estimate the current affective state we used a software tool that analyses a person's facial features and asses the current basic emotion [FR] and modified it to determine the current affective state's valence [BKE+09].

Figure 1 depicts the general architecture of our system. The Context-sensitive Control system is based on a MLP neural network.

For capturing the facial images, we used a high quality web cam and the "FaceReader" affective states assessment software [FR] [BKE+09]. Voice commands and gesture interpretation are considered for future work. The system records the user emotional variations for a specific time period after the system actions are performed. These variations indicate if the system actions were as the user expected or not.

The mechanism works as explained below (see Figure1):

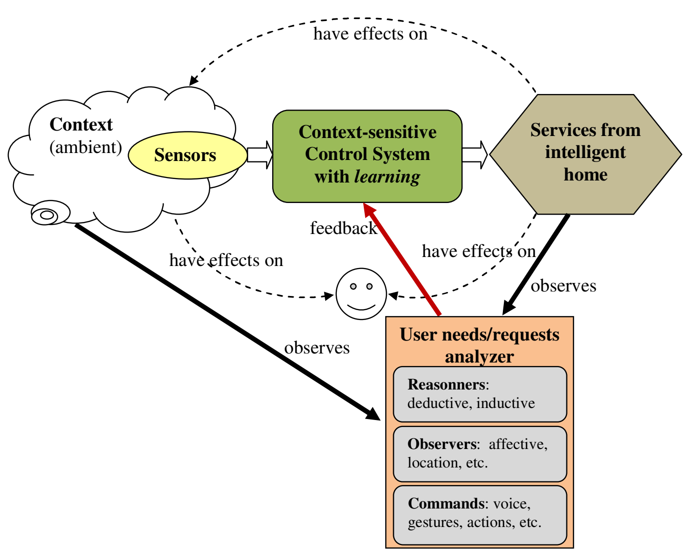

Figure 1: The principle of preference update based on the user's affective response ☺ BAM.

1. At t_0 the system will choose a service for the present context by feeding forward in the randomly or prior trained (with values from that user's behaviour history in similar conditions) neural network.
2. This decision for a service s at t_0 will determine a user reaction in the next time interval t_1. From this reaction, we are interested only in the valence of the emotion: positive (meaning acceptance) or negative (denial).
3. The acceptance or denial will determine the adequate weight w modification. Then the cycle repeats from 1. This way the system adapts itself in successive steps.

The principle for training the neural network is a supervised one. The element that changes during the time is the training set. We used a modified version of the back-propagation algorithm. When the user affective reaction is negative, the desired output is inverted in the training set. The network is re-trained periodically. At this moment, the desired output is estimated only using the emotional reaction but we intend to add also explicit commands and thus the system will learn to respond according to these commands.

4 The Affective-aware Smart Home Implementation

The details of the Affective-aware Smart Home (ASH) implementation are beyond the scope of this paper. Briefly, ASH is based on a Jadex multi-agent system, on Phidgets boards for the sensorial context information gathering and actuators [BHVC+09].

We use ontologies to model the context information because they are independent from any programming language, support formal representation of the context [GPZ05][WZGP04], allow knowledge distribution and reuse, logical context reasoning (consistency check, subsumption reasoning, implicit knowledge inference) [YaL06]. Ontologies provide expressing power (i.e. OWL has cardinality constraints), hierarchical organization, use standards for efficient reasoning, abstract programming and interoperability [ESB07]. By using reasoning mechanisms, the context can be augmented, enriched and synthesized [BMC+06]. Moreover it solves heterogeneity, ambiguity, quality and validity issues related to the context data. [KrS07].

The user related data is usually considered as a part of the context and can also be ontologically modelled [Hec05], including details on her/his affective states [BRC07].

4.1 The Affective Knowledge Representation

We added in the context ontology the State concept as in [BRC07], but, as we were interested by the valence representation for the current state, we defined the subclass CurrentState and for it the valence property with three possible values (positive, negative and neutral) as depicted in Figure 2:

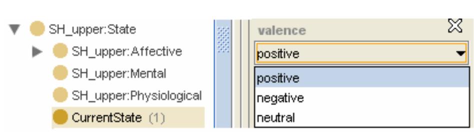

Figure 2: Fragment from the *SH_lower* ontology illustrating a *CurrentState* individual (*left*) and its valence datatype property with the three possible values (*right*)

4.2 Modelling Preferences in a MLP Neural Network

At this stage we implemented our multilayer perceptron (MLP) using a Java that saves the network parameters in a file [BCT09].

The entrance of the MLP had two inputs, the room light (LightSensor1) and the outside light (LightSensor0), with three possible values (low, medium and high) updated into the ontology by the sensor agent:

```
light_indoor=sensorMap.getSensorById("LightSensor1").getSensorV
alue();
```

There is just one output of the neural network, the blinds' status (on/off) that has to be set up in the ontology once a decision is taken:

```
deviceMap.setDeviceStatus("Blind_2","ON");
```

We may compare this with the equivalent Bayes RN Meta-networks [BHVC+09] solution where it is an important increase of prior probabilities with the number of inputs. In our scenario we would have to complete 3^2*2^1=18 combinations, but adding a binary value input (authorized/not authorized user) the number of prior probabilities would double: 36. So, an exponential grows. Moreover the presence of two users demands for one more layer, resulting that for n users n+1 Bayesian layers are needed. As a consequence 3*36 = 108 values need to be computed.

The complexity of the Bayes RN Meta-networks [Anh05] is:

$$O(N*p*q^{\alpha}+q^{\alpha*N*p}) \qquad \mathbf{(1)}$$

Where N is the number of users, p is the user's probability to be in a certain location, q the number of service values or possible actions, α is a value proportional with the number of context elements multiplied by the possible values for that element. For the given example the complexity would be $O\,(1*1*2^6+2^{6*1*1}) = O\,(128)$.

In the neural network case we reduce the complexity to:

$$O(e*q). \qquad \mathbf{(2)}$$

Where e is the number of context elements, q the number of service values or possible actions, so we have O (2*2) =O (4). That reduces the complexity 32 times.

5 Results and Discussions

The experiments were developed in two main stages. The first one was a functional test to see if ASH with ☺BAM is able to learn a new preference after the system was trained with an initial training set [BCT09]. The second stage consisted in simulations of the ☺BAM training when varying the number of context elements. In this second stage we used Weka [HFHP+09] to simulate training the neural network with an initial behaviour preference and then retrain it with a new one. We wanted to see how many training examples have to be provided to the system so that it learns the new preference and we compared this with the to the number of examples needed for the initial behaviour preference learning. In Figure 3, the particular case of four context elements with ternary values considered as inputs of the neural network and one binary service considered as output is depicted. The pink line above shows the initial training phase and the blue one below is the retraining result. The green line, representing 75%

correct behaviour accuracy, is the threshold to consider that the new behaviour is learned in a considerable manner.

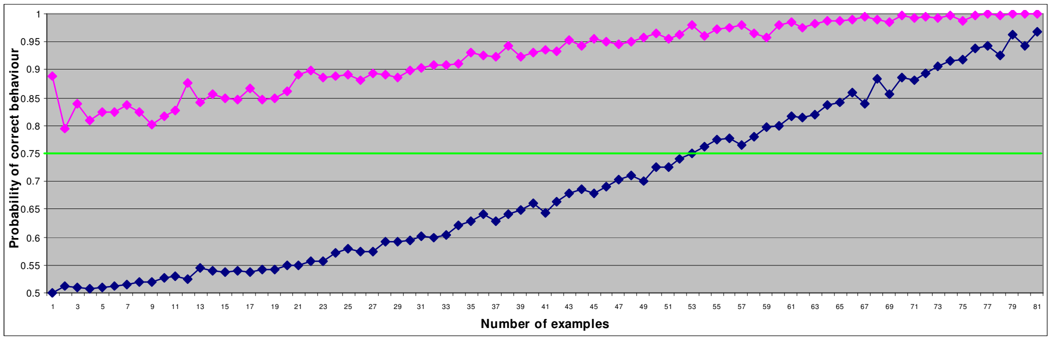

Figure 3: An example of the variation of the number of examples needed for training

We did the same simulations varying the number of context elements from 2 to 10. We noticed that because of the complexity the number of context elements should not exceed 8 otherwise the neural network will not learn from the examples in a consistent manner. However, even a smaller number of context elements is practically problematic, as the number of needed examples increases exponentially like in the Figure 4.

We conclude that some other online machine learning solution should be also selected and tested. In order to do the selection the criteria are:

1. To learn faster the new behaviour preference
2. To allow the retraining even if the number of context elements in big
3. To allow a better generalization (to need for less training examples).

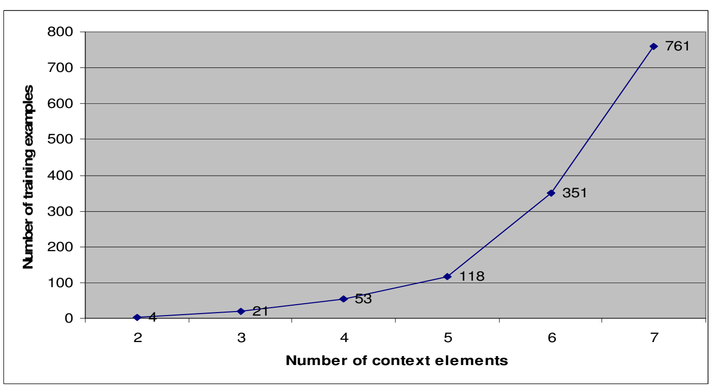

Figure 4: The training examples number needed when the context elements number increases

Acknowledgments

This work was supported by CNCSIS –UEFISCSU, PNII – IDEI, project number 1062/2007.

6 Conclusions and Future Work

We proposed and tested a new behaviour adaptation mechanism, *BAM, for ambient intelligence. This is based on a neural network and is original in the sense that it learns from the user's affective reactions (valence) to the system's decisions. It allows preference discovery, storage and usage for responding to ever changing user needs. At this stage the preferences are stored as neural network weights in a file, but we envision storing them in an ontological representation.

We found out that in order to learn a new behaviour preference, the neural network needs a number of examples that increases exponentially with the number of context elements. That is not practical for a user when more then four context elements are used by the context aware system.

As a future development, we will analyse some other online machine learning solutions to increase robustness to context number increase, reduce the time and number of training instances needed by *BAM to learn the new preference and obtain a better generalisation capability.

References

[ACPP+09] R. Anthony, D. Chen, M. Pelc, M. Persson, M. Törngren. Context-Aware Adaptation in DySCAS. In: *Electronic Communications of the EASST*, vol. 19, 2009, (CAMPUS 2009), Lisbon, Portugal, June 12, 2009.

[ALL05] K. Anh, Y.-K. Lee, S. Lee. OWL-Based User Preference and Behavior Routine Ontology for Ubiquitous System. Pp.1615-1622, LNCS, vol.3761, Springer, 2005.

[Anh05] K. Anh. User Preference Learning in Context-aware Computing. Master thesis, Department of Computer Engineering, Faculty of Graduate School of Kyung Hee University, Korea, 2005.

[BCT09] K.-I. Benţa, M. Cremene, V. Todica. Towards an Affective Aware Home, ICOST'09, LNCS, vol. 5597, pp. 74-81, Springer, 2009

[BHVC09] K.-I. Benţa, A. Hoszu, L. Văcariu, O. Creţ. Agent based smart house platform with affective control, EATIS, Prague, Czech Republic, 3-5 June 2009.

[BKE+09] K.-I. Benţa, H. van Kuilenburg, U. X. Eligio, M. den Uyl, M. Cremene, A. Hoszu and O. Creţ. Evaluation of a System for RealTime Valence Assessment of Spontaneous Facial Expressions. In: Distributed Environments Adaptability, Semantics and Security Issues International Romanian - French Workshop, Cluj-Napoca, Romania, 17-18 Sep. 2009

[BMC+06] L. Buriano, M. Marchetti, F. Carmagnola, F. Cena, C. Gena, I. Torre. The Role of Ontologies in Context-Aware Recommender Systems. In: IEEE International Conference on Mobile Data Management, pp. 80, 2006.

[BRC07] K.-I. Benta, A. Rarau, M. Cremene. Ontology Based Affective Context Representation. In: EATIS, Faro, 2007.

[Bro07] J. Broekens. Emotion and Reinforcement: Affective Facial Expressions Facilitate Robot Learning. LNAI, vol. 4451, pp. 113-132, Springer, 2007.

[CEF09] M. Chan, E. Campo, D. Esteve, J.-Y. Fourniols. Smart Homes - Current features and future perspectives, *Maturitas*, vol. 64(2), pp. 90-97, Oct. 2009.

[ESB07] D. Ejigu, M. Scuturici, L. Brunie. An Ontology-Based Approach to Context Modeling and Reasoning in Pervasive Computing. In: Proceedings of the Fifth IEEE international Conference on Pervasive Computing and Communications Workshops, Washington, 2007.

[Fla05] J. Flanagan: Context awareness in a mobile device: Ontologies versus unsupervised/supervised learning. In: Proceedings of International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning, pp. 167-170, Espoo, 2005.

[FR] Noldus Information Technology, FaceReader™, http://www.noldus.com/human-behavior-research/products/facereader

[GPZ05] T. Gu, H.K. Pung, D. Zhang. A Service-Oriented Middleware for Building Context-Aware Services. Elsevier JNCA, vol. 28(1), pp. 1-18, 2005.

[GYC+07] D. Guan, W. Yuan, S.J. Cho, A. Gavrilov, Y.-K. Lee, S. Lee. Devising a Context Selection-Based Reasoning Engine for Context-Aware Ubiquitous Computing Middleware, LNCS, vol. 4611, pp. 849-857, Springer, 2007.

[HAM+06] Md. K. Hasan, K. Anh, L. Mehedy, Y.K. Lee, S. Lee. Conflict Resolution and Preference Learning in Ubiquitous Environment. In: The 2006 International Conference on Intelligent Computing, LNAI, vol. 4114 0355, Kunming Yunnan Province, 2006.

[Hec05] D. Heckmann. Ubiquitous User Modeling, PhD. Thesis, Saarbrucken, 2005.

[HIR06] K. Henricksen, J. Indulska, A. Rakotonirainy. Using context and preferences to implement self-adapting pervasive computing applications, Software-practice & Experience, vol. 36(11-12), pp. 1307-1330, Ed. John Wiley & Sons Ltd., 2006.

[HFHP+09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. The WEKA Data Mining Software: An Update. In: SIGKDD Explorations, vol. 11(1), 2009.

[KMGA08] R. Kadouche, M. Mokhtari, S. Giroux, B. Abdulrazak. Personalization in Smart Homes for Disabled People, In:*2nd International Conference on Future Generation Communication and Networking,* vol.2, pp.411-415, Dec. 2008.

[KrS07] R. Krummenacher, T. Strang. Ontology-Based Context Modeling. In: 3rd Workshop on Context Awareness for Proactive Systems, Guildford, 2007.

[Moz05] Mozer, M. C. Lessons from an adaptive house. In D. Cook & R. Das (Eds.), *Smart environments: Technologies, protocols, and applications* (pp. 273-294). Hoboken, NJ: J. Wiley & Sons, 2005.

[MPTW09] S. M. McBurney, E. Papadopoulou, N. K. Taylor & M. H. Williams. Implicit Adaptation of User Preferences in Pervasive Systems. In: *Proceedings of the 4th International Conference on Systems : ICONS 2009*, pp. 56-62. IEEE Computer Society. Cancun MEXICO, 2009.

[MTWP09] S. McBurney, N. Taylor, H. Williams, E. Papadopoulou. Giving the User Explicit Control over Implicit Personalisation. In Proc. Persist Workshop on Intelligent Pervasive Environments (AISB 09), (in press), 2009.

[NYS+05] K. Nishigaki, K. Yasumoto, N. Shibata, M. Ito, T. Higashino. Framework and rule-based language for facilitating context-aware computing using information appliances. In: First International Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet, pp. 345-351, 2005.

[RaC09] P. Rashidi and D. Cook. Keeping the resident in the loop: Adapting the smart home to the user. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 39(5), pp. 949-959, 2009.

[SHRS08] J. Schmitt, M. Hollick, C. Roos, R. Steinmetz. Adapting the User Context in Realtime: Tailoring Online Machine Learning Algorithms to Ambient Computing. MONET 13(6), pp. 583-598, 2008.

[SKW05] Y. Suh, D. Kang, W. Woo. Context-based User Profile Management for Personalized Services, ubiComp, workshop ubiPCMM, 2005.

[WZGP04] X. Wang, D. Zhang, T. Gu, H. K. Pung. Ontology Based Context Modeling and Reasoning using OWL. In: Proceedings of CoMoRea, Orlando, 2004.

[YaL06] S. S. Yau, J. Liu. Hierarchical Situation Modeling and Reasoning for Pervasive Computing. In: Proceedings of the Fourth IEEE Workshop on Software Technologies For Future Embedded and Ubiquitous Systems, and the Second international Workshop on Collaborative Computing, integration, and Assurance, Washington, 2006.