Challenges in Case-Based Reasoning for Context Awareness in Ambient Intelligent Systems

Anders Kofod-Petersen

Department of Computer and Information Science,
Norwegian University of Science and Technology,
7491 Trondheim, Norway
anderpe@idi.ntnu.no,
http://www.idi.ntnu.no/

Abstract. One of the most important issues in an ambient intelligent environment, indeed in any problem-solving situation, is the ability of a system to appreciate its environment and assess the situation in which problems are to be solved. Recently, case-based reasoning has gained some momentum in the area of context awareness for ambient intelligent systems. When applying case-based reasoning in this type of system some challenges arises, each well known for case-based reasoning in general, but each gaining a new particular angle. The main challenges are: how to acquire the initial cases, coping with the potential for a very large amount of cases, when to execute a case-based reasoning process, and knowing if the reasoning was correct. The work presented here builds on the experiences gained by applying case-based reasoning to situations assessment. It will analyse each of these challenges from the perspective of different domains, and give some suggestions as to how to best approach these challenges.

1 Introduction

One the most important aspects when achieving an ambient intelligent environment, is the ability to identify the situation occurring. Standard software applications implicitly "know" the situations in which they are to function. When a developer constructs a piece of software he knows the specific situations where it is to be used. In other words, he knows which inputs are used to what and why. If we examine applications that are to function within an ambient intelligent environment, the ability to know exactly what should be done, when, how and why are not necessarily well know when constructing an application. A well-known example of envisioned functionality of this kind is the common sense reasoning, adaptation, and pro-activity displayed by the Aura system [1].

Context awareness is the term that is most often used to describe the ability of an application to adapt to these unknowns. Even though a lot of research has been done within the field of context awareness, no common definition of what the term encompasses has been agreed upon. Further, it is not clear whether such terms as context aware, context sensitive or situation aware are synonymous, or they are somehow sub-sets of each other. However, it is commonly agreed the context awareness allows applications to dynamically adapt to the environment.

Up until now, most of the work on context aware systems has been largely technology driven, often "... driven by what is technological feasible rather than by what might be helpful in a situation." [2, p.1] This approach has lead to a number of applications that hardly can be described as *aware*. Rather, they are *sensitive* to context, often as classical *stimuli-response* based applications.

We argue that the ability to be aware of the context and assess situations is at the core of an ambient intelligent system. Our hypothesis is that case-based reasoning supported by a rich knowledge model, is a promising approach to asses situations by being context aware [3]. However, using case-based reasoning in an online fashion in a inherently dynamic environment poses a number of challenges. We have identified four main challenges. These are not a function of using case-based reasoning within ambient intelligence, yet the severity and complexity increases. These are:

1. Acquiring the initial cases
2. Coping with the vast number of case being constructed during run time
3. Knowing when to initiate a case-based reasoning cycle
4. Knowing whether a case was classified correctly

This work will analyse these four main challenges in the light of our experiences from developing an ambient intelligent system utilising case-based reasoning. The rest of the paper is organised as follows: First a short overview of use of case-based reasoning for context awareness is given. Secondly, the background of this work is described. This is followed by a description of the domains we have been investigating, how the challenges were manifested, and our suggestions on how to approach them. Finally, a summary of the approaches suggested is given, as well as an outlook on future work.

2 Use of Case-Based Reasoning in Ambient Intelligence

Case-based reasoning has been utilised as a method for identifying situations in a dynamic environment. Zimmermann [4] demonstrated how case-based reasoning could be used to generate recommendations for users in a mobile environment. The recommendations were used to augment a museum exhibition with audio information. Zimmermann recognised the problem of a rapid growth in the number of case in a case-base. To solve this problem, he proposed to initially cluster similar cases and consecutively generalise cases prior to the reasoning process.

Kwon and Sadeh [5] applied case-based reasoning to find a *pareto* agreement between a consumer and seller in comparative shopping. The approach presented by Kwon and Sadeh was a simulation to compare algorithmic efficiency, and did therefore not explicitly address any of the challenges aforementioned.

Ma et al. [6] used case-based reasoning to adapt user preferences in smart homes. The system presented addresses the issue of acquiring initial case by simply observing the settings of a particular user. The question of whether or

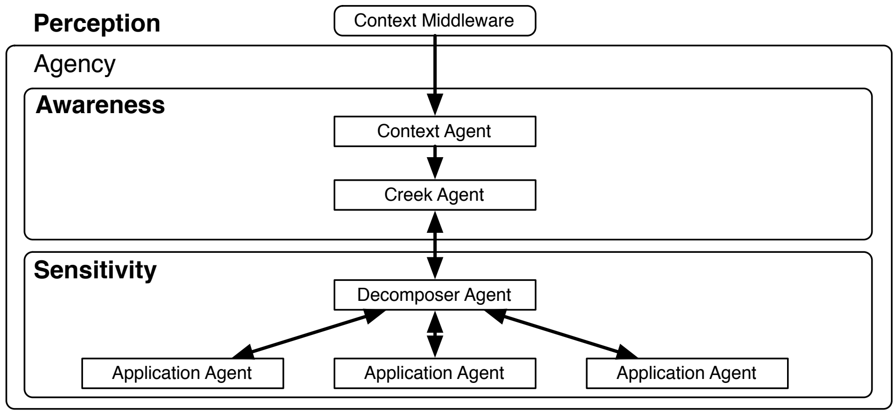

Fig. 1. Structured System Architecture

not a proposed solution was correct or incorrect was measured by observing how a user interacted with the system. The interaction was sensed through the ordinary user interface to a house: switches, remotes, etc.

The MyCampus [7] reports on the use of case-based reasoning to learn a user's context-sensitive message filtering preferences. Cases were acquired by storing the user feedback along with the messages received. This was also the approach to learning in the case-based, by user feedback.

Bénard et al. [8] describes a framework that uses case-based reasoning to identify the most suited action to perform. The choice of action is based on a comparison between perceived information and earlier experiences.

3 Background and Motivation

The foundation for the analysis and suggestions made in this paper is based on the work carried out through the last four years. The aim has been to construct a system that can be context aware, or perceive the world and through the knowledge intensive case-based reasoning in CREEK [9] assess situations. The system has a three layered architecture (see Figure: 1), where each layer has its own specific function.

The Perception layer uses the Context Middleware [10] to perceive the world and order the observations into a coherent structure useable for the rest of the system. The data perceived can in theory be any data that can be gathered from sensors, both from the physical world, but also from the computerised world. The Awareness layer is responsible for assessing the context and classify the situations through the case-based reasoning cycle [3]. Finally, the Sensitivity layer infers what sequence of actions are sensible in a give situation and executes them if possible [11].

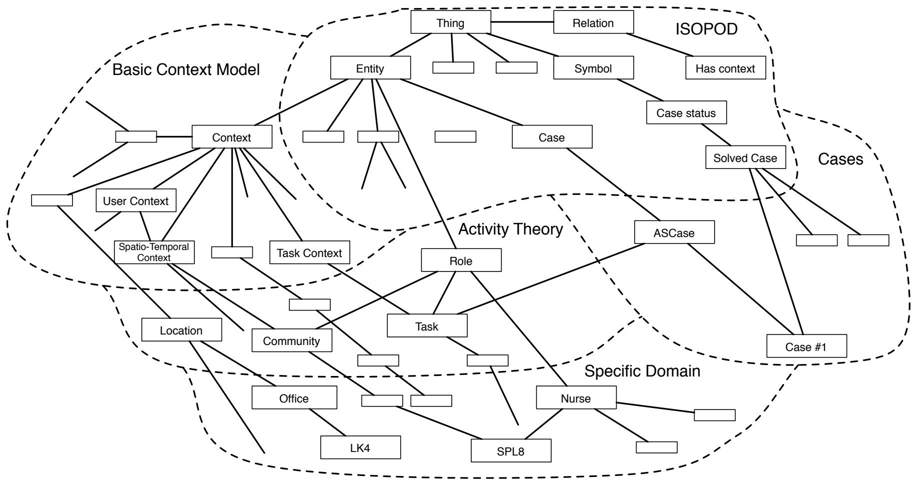

Fig. 2. Overview of the Knowledge Model

The whole system assumes a knowledge intensive approach to context awareness. The knowledge model is considered from a knowledge level [12] perspective, and modelled using the socio-technical perspective gained through applying Activity Theory [13]. The knowledge model is structured into five main parts, see Figure 2. The top level otology, acquired from the CREEK system, called ISOPOD; the basic context model, which is a meronomy building on tradition in the area of pervasive computing; the part that deals with activities, which is founded on Activity Theory; the concepts that are specific to the domain in question; and finally, the cases that describes the situations. The whole model is represented as a multi-relational semantic network.

Context, as a term, is used used in two slightly different meanings. During execution of the system, context is initially used to describe the subset of the world that can be perceived, through the Context Middleware. This context is what the awareness part can used to described the situation in question. This context is stored in cases as *findings*. The second use of the term context deals with the information available to the system when solving the problem in the situation. In the latter use, the context is a subset of the perceived world, which has been focused by the assessment process. Thus, leaving out information which is unimportant for the tasks that are to be executed to satisfy the problem.

The system was originally applied to the tourist domain in an effort to supply contextualised and personalised information to travellers [10]. It has later been applied to the health care in a hospital ward domain, where situation classification was used to acquired relevant information for the staff at a hospital ward [3,14]. We are currently investigating whether the system is applicable in a mobile collaborative learning domain [15], and we are also revisiting the original use in a tourist domain [16].

Through the work with case-based reasoning as the main reasoning process in context awareness, we have encountered the four main challenges in the domains we have investigated. The challenges have basically been the same across domains, yet each domain has had its own particular twist to the challenges.

4 Tourist Domain

The tourist domain is characterised by the way the case-based reasoning cycle is used to look at the current situation to retrieve a recommendation for future actions, which would be beneficial for the user to execute. The recommendations are not only based on what information is available in the environment, but also to a large degree on the preferences of the user in question. An example is the situation where the case-based reasoning system recognises that the user might be hungry and suggests a course of actions that will bring the user into a restaurant that fits his preferences [17]. In this example, a new case's *findings* contains the information that can be perceived, such as: the user's preferences, the location, time, and any entities accessible in the environment.

The application of a case-based reasoning based system —indeed any context aware system— into this type of scenario leads to specific versions of the four challenges described in the introduction:

Initial, the problem is to acquire the initial cases prior to executing the system, known as prototypical case. Given that applications in this domains aims at delivering personalised information services, a focus on personal preferences is required. These preferences are trivial to acquire and can be used to construct user stereotype, in the tradition of Rich [18]. Along with the domain model, these stereotypes is the foundation for modelling the context-goal combinations. Regarding the context example described above, a prototypical case contains the findings that a user prefers a particular type of restaurant, that he is located near such a restaurant, and that the time of day is around lunch. The *solution* contains the tasks that must be fulfilled in order to guide the user to a suitable restaurant, such as finding a suitable restaurant in the vicinity, and guide the user to that location. Even though these cases would only be partly personalised, they will make a good foundation for adapting to a particular user's idiosyncrasies through learning in the case-based reasoning cycle.

The second problem is to maintain the potential very large case base. When dealing with a case-based reasoning system running in an on-line manner, there is a real risk of storing a very high number of cases. One part of this issue springs from the fact that some of the parameters observed are continuous, such as time. Since a case is a discrete point in a multidimensional space, the continuous parameters must be made discrete. Thus, leading us to the question of how often does time trigger a new case-based reasoning cycle, without constructing to many cases, yet still ensuring that the system covers significant changes in the environment.

Initially, there are two approaches to this challenge: either the system should be able to forget cases that are not useful [22,23]; or some sort of off-line generalisation should occur, such as clustering of similar cases [4] or generalisation into prototypical cases [24].

The third challenge ties in with the problem of many cases. To avoid an overwhelming number of cases, it is essential to decide when to instantiate a new case. Two main approaches to this issue stand out. The first is to employ some type of statistical, or pattern matching algorithm to figure out whether a new observation is so dissimilar that a new case-based reasoning cycle is warranted; or simply follow the example of Fox et al. who constructed a case-based reasoning controller for a mobile robot [19]. They initially suggested looking at how to weight different parameters present in the environment. However, it became clear that the case-based reasoning cycle was fast enough, so they simply observed the environment every seventh second. This simple solution obviously assumes that the execution of the cycle is both computationally and commercially inexpensive.

The fourth challenge is to figure out whether the goals associated with a case, or the tasks suggested for a particular situation are correct; or in other words, guiding the retain step. If we revisit the example above where the system identifies that the user is likely to be hungry, and a visit to an Italian restaurant is the solution, we see that the system will not instantly known if this suggestion was good. There are many factors that might influence the user's behaviour, which might not be captured by the system's knowledge. The user might disregard the suggestion because he suddenly feels like Chinese, he might not be hungry at all, or he simply doesn't have the time right now.

There are basically two approaches for solving this: simply asking the user, or observe the behaviour of the user and see if it fit the (implicit) projection made by the system. In this case it is likely to be unfeasible to ask the user: "was this a good suggestion?" By asking the user, the system runs the risk of being to intrusive, continuously asking question, leading to the user turning off the annoying thing. Of course, the amount of conversation could be kept to a minimum by applying conversational case-based reasoning techniques as suggested in [20]. However, it is likely to be feasible to look at the behaviour of the user over time, and through these observations attempt to recognise whether or not the suggestion made was any good.

When the system makes a suggestion to the user, it also, implicitly, makes a prediction on the user's behaviour in the (near) future. As time progresses and new instances of the user's context, or cases, is acquired by the system, it is possible to see whether a case that verifies or falsifies the prediction occurs. This analysis of the cases being instantiated, requires some way of representing temporal knowledge in the case base, and the ability to make predictions [21]. In the examples above, it is easy to verify that the suggestion made was good. If a situation that describes the user having lunch at the Italian restaurant occurs, is it reasonably to conclude that the suggestion was good, and retain the original case as a positive example.

Table 1. Challenges and Solutions for the Tourist Domain

Challenges	Solutions	
Acquiring initial cases	User modelling through stereotypes	
Maintaining case based	Forgetting	Clustering/Generalisation
Initiating CBR-cycle	Initial matching	Regular execution
Guiding Retain	Asking the user	Observing behaviour

Figuring out whether the suggestion made was wrong is more complicated. There are two ways of observing that the suggestion made was wrong: the user behaves in a way that explicitly contradicts the predicted behaviour, by for an example visiting a Chinese restaurant; or the user behaves in a way where it is not certain that he follows the suggestion made, such as behaving in a way that neither contradicted nor fulfilled the prediction within a reasonably time frame.

Table 1 gives a summary of the challenges in the tourist domain, and the solutions suggested. Except for the issue of acquiring the initial cases, all challenges have (at least) two different approaches for solutions, each offering their own benefits and pitfalls. Regarding how to maintain the case base, the choice between forgetting cases that are no longer relevant, versus doing of-line generalisation is very likely a question of the diversity in the situations observed. When to initiate a full case-based reasoning cycle, the question of doing an initial match to the present situation or simply reclassify, seems to be depending on how expensive the execution is. Finally, deciding how to retain a case is either by asking the user or observing his behaviour, where the choice is likely to be guided by how annoyed the user gets by being asked. However, it is important to note that none of the suggestions above are mutually exclusive and some combinations are likely to be the most feasible.

5 Health Care in a Hospital Ward Domain

The use of case-based reasoning for situation assessment in health care is somewhat different from the use in the tourist domain. Our use of case-based reasoning is in hospital wards, where the intention is to recognise situations and execute actions that will gather information, which is relevant for the health care professionals' workflow [3,14]. This is different to the tourist domain, where the objective is to suggest actions. Where the tourist domain is very dynamic with regards to where situations occur, there is a high degree of correlation, even to some degree causality, between the location and/or time of day, and the type of situation in a hospital ward.

The use of our case-based reasoning system in this setting perceives the environment and attempts to classify situations. Based on this classification the system will execute a series of actions, which will acquire the information required in the work process in this particular situation. As with the tourist domain, the findings of the cases here contains the information that are perceived, such as:

location, persons present, artefacts available, and time of day. The solution part of the cases contains the sequence of tasks, of which the sensitivity layer must execute corresponding actions.

One example is when the system recognises that a meeting of the type *pre ward round* is occurring. A *pre ward round* is a type of meeting that occurs every morning in the ward. In this meeting the patients who are to be visited later during the ward round are discussed. The typical chain of tasks is:

1. Acquire name of patient
2. Acquire changes in patient's conditions since yesterday
3. Examine, and possible change, medication scheme
4. Acquire any new results from tests
5. Note changes in treatment

The information required for each of these steps can typically be acquired from any number of sources, both artefacts and people. The problem lies in recognising an ongoing situations as a *pre ward round*, and acquire the information required from the source that are present and able. For the particular *pre ward round*, the different artefacts and persons supplying the information is [3]:

1. Patient list
2. Nurse
3. Patient chart
4. Patient chart, electronic patient record
5. Nurse

Revisiting the four challenges described in the introduction in the light of applying case-based reasoning in a hospital ward, leads to specific versions of these challenges:

To acquiring the initial cases and the workflow descriptions, we have opted for a model where we observe existing situations and workflow. The observations where based on a socio-technical analysis of the domain as described in [13,14]. This domain is to a large degree governed by procedures, and the main problem in each of the situations is to figure out the sequence of tasks, and the information supplied by what artefact and people. The personalisation issues, regarding a user's preferences, where not important. In other words, each case that describes a type of situation is build through thorough observation, rather than constructing simple cases based on user models in the tourist domain.

Secondly, the question of a large amount of cases exists. In the tourist domain, there was a potential for a very large amount of cases. However, as described above the hospital domain has a higher degree of causality, thus instantiating new cases should happen less frequent. Also, the higher degree of predictability due to procedures should result in a less diverse case base; i.e. less information gain in each new case. All in all, where forgetting might be a good approach in the tourist domain, it seems less useful in a hospital ward. Whereas, the use of clustering, or generalisation into prototypical cases is much more feasible.

Table 2. Challenges and Solutions for the Health Care Domain

Challenges	Solutions	
Acquiring initial cases	Observations of existing situations	
Maintaining case base	Clustering/Generalisation	
Initiating CBR-cycle	Change in primary parameter	Regular execution
Guiding Retain	Asking the user	Observing behaviour

The third challenge is to know when it is feasible to execute a full case-based reasoning cycle. Again, this challenge is primarily based on the assumption that it is either computationally of commercially expensive to execute a full cycle. For the hospital ward domain, it is safe to assume that commercial issues do not affect execution. Thus, only the question of computational cost is important. As already stated there is a strong relationship between the location and/or time of day, and the type of situation occurring. Given these premises, the choice of whether or not to execute a full case-based reasoning cycle can often be made based on the location of the user and/or the time of day with regard to the ward's schedule. However, given the results by Fox et al. described in the tourist domain, it is also likely that a simple regular execution is feasible.

Finally, once the case-based reasoning system is running, one of the main challenges is to decide whether the goal associated with a situation is correct. As with the tourist domain, there are two obvious approaches: ask the user, or observe the behaviour. As for the tourist domain it is unlikely a good idea to have the "ask the user" strategy as the sole approach; rather some form of observation of behaviour of the user and environment is preferable.

In the tourist domain, a considerable amount of time might separate the time when the system suggests a course of actions, and the point where it is possible to recognise whether the suggestion was sound. This is not such a big problem in the hospital ward. As mentioned above, the situations occurring and the task executed in each situation is much more structured in a hospital ward. Thus, we generally have a much shorter time span between suggestion and effect. This is also true for the types of behaviour that should be observed. As the main problem is to acquire relevant information for the user, the main observation is simply monitoring what types of information, and from what artefacts, the user accesses. If for an example the user chooses to acquire new results (point 3 in the example above) from another artefact that the one suggested, it is obvious that the system must update its model of the artefacts. If, on the other hand, the users accesses a new type of information from an existing or new artefact, the system needs to update its workflow model. Finally, the user might behave in a totally different way than predicted, thus the system must reclassify the existing case.

Table 2 give a summary of the suggested approaches to the four challenges when applying case-based reasoning in a hospital ward. Regarding when to execute a case-based reasoning cycle, it seems that the choice is between monitoring a limited number of key parameters, or simply executing at fixed intervals. When deciding how to guide the retain phase, it is for the hospital domain easier to monitor the behaviour, thus asking the user should not be necessary too often. Finally, as the diversity in the situations occurring is limited, a clustering approach seems to be the most suitable.

6 Summary and Further Work

This work has build upon the experiences gained through working with case-based reasoning for situation assessment in an ambient intelligent setting. We have developed a system and applied it in simulation to both a tourist domain and a health care in a hospital ward domain. Through our experiences we have identified four challenges when applying case-based reasoning. Each of these challenges are by no way unique to the ambient intelligence area, however each is influenced by the nature of situated and real time case-based reasoning.

If we compare the suggestions for dealing with the challenges across the two domains, we can see that even though each domain has its own idiosyncrasies, the approaches suggested are similar. Table 3 described the suggestions in each of the domains, where the tourist domain is the first.

The area where the domains differ the most is in acquiring the initial cases. In the tourist domain the aim is to supply localised and personalised information in run time. This assumes a good model of the user and his preferences. These preferences could of course be learned overtime. However as we are building software that are to be used by a human user, it is preferable that the learning period is a short as possible. Therefore, an approach using stereotype models is suggested. This is in contrast to the hospital ward, where situations and workflow have a much higher degree of organisation. The personal preferences of a user do not affect the situation nearly as much as when he is a tourist. Thus, a thorough modelling of the domain through observations is suggested.

Regarding when to initiate a case-based reasoning cycle, it is not clear whether this is a problem at all. In particular, it seems reasonably to assume that in the hospital domain a regular execution of the case-based reasoning cycle it the sensible solution. However, it is less obvious whether this is the solution for the tourist domain. It is presumably better to do some pre-processing of the sensory input, to decide whether the new input is significantly different from the current situation, thus limiting the number of executions of case-based reasoning cycle.

When we look at how to guide the retain process, the main balance is between not being intrusive by observing behaviour, and getting first hand knowledge by asking the user. In the tourist domain there can be a significant time span in which behaviour needs to be observed, leading to several problems when deciding whether a particular suggestion was right or wrong. In this case it can be hard to weigh the two approaches. In the hospital domain, the time frame for each observation is much smaller, thus is would seem to be beneficial to primarily observe behaviour.

Table 3. Summary of Challenges and Solutions

Challenges	**Solutions**	
Acquiring initial cases	User modelling through stereotypes Observations of existing situations	
Maintaining case based	Forgetting Clustering/Generalisation	Clustering/Generalisation
Initiating CBR-cycle	Initial matching Change in primary parameter	Regular execution Regular execution
Guiding Retain	Asking the user Asking the user	Observing behaviour Observing behaviour

Finally, maintaining the case base is one of the issues that so far have been ignored. No clear advantage of either forgetting or clustering has been identified. Even though there seems to be an indication that forgetting can be beneficial in the tourist domain, whereas clustering is most suited to domains with the characteristics of hospital wards.

This is ongoing work in both the tourist and hospital domain, and in that way no clear conclusions are available. However, it seems clear that both domains share the same challenges, yet each must be approach in a slightly different way. Thus, the main conclusion is that there is no such thing as a “one size fits all”.

We are currently investigating the applicability of our approach in the mobile collaborative learner domain [15]. Initially, we assume that this domain resembles the tourist domain in many ways, thus the approaches suggested for the tourist domain should be applicable for the mobile collaborative learner domain. However, we expect that the idiosyncrasies of this particular domain will once again tell us that “one size fits nobody”.

References

1. Satyanarayanan, M.: Pervasive computing: Vision and challenges. IEEE Personal Communications **8** (2001) 10–17
2. Lueg, C.: Operationalizing Context in Context-Aware Artifacts: Benefits and Pitfalls. Information Science – The International Journal of an Emerging Discipline **5** (2002) 42–47
3. Kofod-Petersen, A., Aamodt, A.: Contextualised ambient intelligence through case-based reasoning. In: Proceedings of the Eighth European Conference on Case-Based Reasoning (ECCBR 2006). Lecture Notes in Computer Science, Ölüdeniz, Turkey, Springer Verlag (2006) To appear.
4. Zimmermann, A.: Context-awareness in user modelling: Requirements analysis for a case-based reasoning application. In Ashley, K.D., Bridge, D.G., eds.: ICCBR 2003, Case-Based Reasoning Research and Development. Number 2689 in Lecture Notes in Artificial Intelligence, Springer-Verlag (2003) 718–732
5. Kwon, O.B., Sadeh, N.: Applying case-based reasoning and multi-agent intelligent system to context-aware comparative shopping. Decision Support Systems **37** (2004) 199–213

6. Ma, T., Kim, Y.D., Ma, Q., Tang, M., Zhou, W.: Context-aware implementation based on cbr for smart home. In: Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE, IEEE Computer Society (2005) 112–115
7. Sadeh, N., Gandon, F., Kwon, O.B.: Ambient Intelligence: The MyCampus Experience. Technical Report CMU-ISRI-05-123, Carnegie Mellon University (2005)
8. Bénard, R., Bossard, C., Loor, P.D.: Context's modelling for participative simulation. In: Proceedings of te 19th FLAIRS Conference, AAAI Press (2006) 613–618
9. Aamodt, A.: Knowledge-intensive case-based reasoning in creek. In Funk, P., Calero, P.A.G., eds.: Advances in case-based reasoning, 7th European Conference, ECCBR 2004, Proceedings. (2004) 1–15
10. Kofod-Petersen, A., Mikalsen, M.: Context: Representation and Reasoning – Representing and Reasoning about Context in a Mobile Environment. Revue d'Intelligence Artificielle **19** (2005) 479–498
11. Gundersen, O.E., Kofod-Petersen, A.: Multiagent Based Problem-solving in a Mobile Environment. In Coward, E., ed.: Norsk Informatikkonferance 2005, NIK 2005, Institutt for Informatikk, Universitetet i Bergen (2005) 7–18
12. Newell, A.: The Knowledge Level. Artificial Intelligence **18** (1982) 87–127
13. Kofod-Petersen, A., Cassens, J.: Using activity theory to model context awareness. In Roth-Berghofer, T.R., Schulz, S., Leake, D.B., eds.: Modeling and Retrieval of Context: Second International Workshop, MRC 2005, Revised Selected Papers. Volume 3946 of Lecture Notes in Computer Science., Edinburgh, UK, Springer Verlag (2006) 1–17
14. Cassens, J., Kofod-Petersen, A.: Using activity theory to model context awareness: a qualitative case study. In: Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference, Florida, USA, AAAI Press (2006) 619–624
15. Petersen, S.A., Kofod-Petersen, A.: Learning in the city: Context for communities and collaborative learning. In: Proceedings of the 2nd International Conference on Intelligent Environments (IE06), Athens, Greece, The Institution of Engineering and Technology. (2006) To appear.
16. Petersen, S.A., Kofod-Petersen, A.: The non-accidental tourist: Using ambient intelligence for enhancing tourist experiences. In: Proceedings of the 7th IFIP Working Conference on Virtual Enterprises, Helsinki, Finland, Springer Verlag (2006) To appear.
17. Kofod-Petersen, A., Aamodt, A.: Case-Based Situation Assessment in a Mobile Context-Aware System. In Krüger, A., Malaka, R., eds.: Artificial Intelligence in Mobile Systems 2003 (AIMS), Universität des Saarlandes (2003) 41–49
18. Rich, E.: Stereotypes and user modeling. In Kobsa, A., Wahlster, W., eds.: User Models in Dialog Systems. Springer Verlag (1989) 35–51
19. Fox, S.E., Anderson-Sprecher, P.: Robot navigation using integrated retrieval of behaviours and routes. In: Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference, Florida, USA, AAAI Press (2006) 346–351
20. Gu, M., Aamodt, A.: Dialog learning in conversational cbr. In: Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference, Florida, USA, AAAI Press (2006) 358–363
21. Jære, M.D., Aamodt, A., Skalle, P.: Representing temporal knowledge for case-based prediction. In: Advances in case-based reasoning. 6th European Conference, ECCBR 2002, Lecture Notes in Artificial Intelligence, 2416, Springer Verlag (2002) 174–188

22. Smyth, B., Keane, M.T.: Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada (1995) 377–383
23. Kira, Z., Arkin, R.C.: Forgetting bad behavior: Memory management for case-based navigation. In: Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Volume 4., IEEE (2004) 3145–3152
24. Schmidt, R., Gierl, L.: The roles of prototypes in medical case-based reasoning systems. In Burkhard, H.D., Lenz, M., eds.: 4th German Workshop on CBR – System Development and Evaluation, Humbolt University, Informatik-Berichte, Berlin (1996) 207–216