Contextualised Ambient Intelligence through Case-Based Reasoning

Anders Kofod-Petersen and Agnar Aamodt

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

Abstract. Ambient Intelligence is a research area that has gained a lot of attention in recent years. One of the most important issues for ambient intelligent systems is to perceive the environment and assess occurring situations, thus allowing systems to behave intelligently. As the ambient intelligence area has been largely technology driven, the abilities of systems to understand their surroundings have largely been ignored. This work demonstrates the first steps towards an ambient intelligent system, which is able to appreciate the environment and reason about occurring situations. This situation awareness is achieved through knowledge intensive case-based reasoning.

1 Background and Motivation

Mark Weiser coined the term ubiquitous computing to describe the way computers would "...weave themselves into the fabric of everyday life until they are indistinguishable from it" [1, p. 1]. Since then this area of research has gained considerable impetus, as well as many names. Although there are subtle differences, the term pervasive computing is also frequently used to refer to these kind systems. Methods that aim to give this type of highly distributed systems intelligent properties, are often referred to by the label *ambient intelligence*.

The visions and scenarios described within the wide area of pervasive computing have grown from the original and simple, describing augmented artefacts assisting users in their day to day living, into full grown systems that assume responsibility from the user, and display a large degree of common sense reasoning as well as elaborate problem-solving. Examples of these ambient intelligent systems can be seen in the way Fred receives help from the omnipresent system Aura [2, p. 3], or in the way Maria is helped through her business trip [3, p. 4].

To display these kinds of complicated behaviours, an entity must be able to interpret the environment in which it is situated, i.e. possess a sufficient level of *context-awareness*. So far, most of the work on context-aware computing have been largely technology driven, leading to, for example, a large number of ways to identify the location of individuals or artefacts using various kinds of position detectors. Hence, many of the pervasive computing systems around today

are rarely more than stimuli-response systems, which regard context merely as location, as earlier pointed out in [4]. Over the last few years, however, there has been a growing interest, within pervasive computing in general and ambient intelligence in particular, in issues surrounding the modelling and representation of context in a wider sense [5,6].

A broader view on context, and an active role for context in situation assessment, calls for an explicit model of context that is an integral part of the overall knowledge model. Our hypothesis is that case-based reasoning (CBR), supported by a rich model of general domain knowledge, is a promising way to achieve ambient intelligent systems. The very nature of CBR lends itself easily to reasoning about context and situation assessment. The use of context to guide the case-based reasoning process has been demonstrated in [7], where *relevance* and *focus* are regarded as the essential properties of context. The goal of an agent becomes part of the context used to focus the attention, and thereby to identify the knowledge needed to execute the actions associated with the goal.

Zimmermann demonstrates the feasibility of CBR for identifying the correct combination of parameters required to display situation understanding [8]. He proposes to initially cluster similar cases, and consecutively generalise cases prior to the reasoning process. The work presents an instance-based approach to the underlying problem of all context aware pervasive computing applications; namely to identify similarities between contextual parameters, using a general reasoning mechanism.

In the work by Ma et al. [9] case-based reasoning is used to adapt the behaviour of smart homes to users' preferences. A multi-user smart home can, even with a very limited number of connected devices, present itself with a large number of interdependent processes. CBR is used to identify these interdependencies, due to its ability to reason in ill-defined and poorly structured domains. Cases are represented as frames, where the findings are: the user, the environment, the time, and the values from active devices. When new cases are instantiated, the similarities between the new and existing cases are calculated using 1- or k-nearest neighbour.

Kwon and Sadeh [10] applies CBR to a multi-agent environment in order to estimate a best purchase in comparative shopping. In comparative shopping consumers who wish to purchase a product can compare prices, warranties, and other aspects between suppliers of a product. Correspondingly, suppliers can tailor their products to specific customers or customer groups. Since it is difficult to estimate the buyers' exact utility function, a negotiation agent employs case-based reasoning to match a current state to possible outcomes. The findings of a case are the product in question, the price, the level of quality, together with contextual information consisting of location, weather and calendar info.

The above methods have clearly shown the potential of CBR for ambient intelligence. However, they are all characterised by a k-NN, or knowledge-poor, approach to similarity assessment, and no explicit or elaborated model of context exists. Hence, reasoning about context and the role of context in situation assessment, at different levels of abstraction, is problematic.
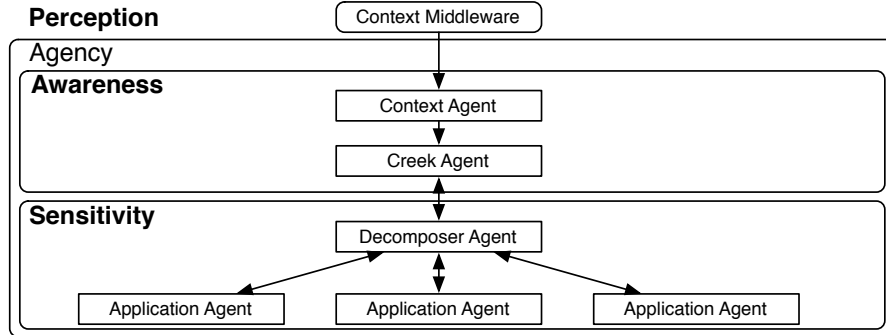
Perception — Context Middleware

Agency

Awareness

Context Agent

Creek Agent

Sensitivity

Decomposer Agent

Application Agent | Application Agent | Application Agent

Fig. 1. Structured System Architecture

We are studying how knowledge intensive CBR can be a suitable method for achieving situation awareness. To obtain this, careful considerations must be made when defining the knowledge. We argue that our particular approach to CBR, in which cases are submerged within a general knowledge model, allows for elaborate reasoning about context beyond what is obtainable in other approaches. In our work we lean on results from within the pervasive computing community, combined with insight from Activity Theory [11]. Our approach also draws on results from work on cooperative CBR [12] as well as ontology modelling of CBR processes [13].

The work described here started within the EU project called AmbieSense, aimed at providing focused guidance and targeted information pushing for people on travel [14,15]. Rather than travelers, our current example application is for support of health workers cooperating in patient diagnosis and treatment. Individual PDA-based assistants will provide information based on their assessment of the current situation, which also includes being able to agree on a common interpretation of the current situation. Our results so far do not demonstrate a final solution for achieving ambient intelligence; rather they show how the combination of our particular knowledge model, the description of cases, and the system architecture, is a promising approach to achieving ambient intelligence.

The paper is structured as follows: First, a short overview of the system architecture is given. Secondly, the knowledge model used to describe the environment, situations and cases is explained. Thirdly, the case model is described. This is followed by an example in our domain of situation assessment within a hospital ward. A conclusion and pointers to future work ends the paper.

2 System Architecture

The system is designed as a three layered architecture (see Figure 1). Each of the three layers has its own specific responsibility. This three layered division is
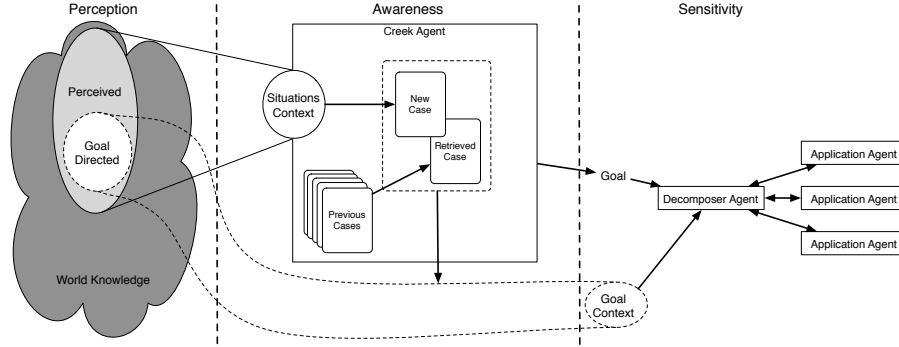
Fig. 2. Functional System Architecture

comparable to the three levels required to achieve *situation awareness* proposed by Endsley et al. [16, p. 14], where there is a *perception* layer, corresponding to our Perception layer; a *comprehension* layer, comparable to our Awareness layer; and a *projection* layer, which roughly corresponds to our Sensitivity layer.

The *Context Middleware* layer in Figure 1 provides a generic context management infrastructure that collects and maintains data from the environment, in a coherent and structured way [15]. This layer communicates with the agency through the *Context Agent*, which receives notifications of changes in the context. This notification is translated into the ontology used inside the agency, and relayed to the *CREEK Agent*.

The CREEK agent is the agent responsible for assessing the context and classifying the situation. This agent uses and extends the CREEK CBR system [17]. Once a classification of the situation has been achieved, the goal associated with the specific type of situation, and the context describing the part of the world available for satisfying the goal are transmitted to the *Decomposer Agent*.

The decomposer agent is responsible for selecting the correct sequence of tasks for execution, as well as drafting the correct *Application Agents* necessary. The latter is done by querying a matchmaker agent for agents having the correct post- and pre-conditions for execution of the sub-tasks. UPML [18] is used to describe the components of plans, corresponding to task sequences. Once a suitable collection of agents has been gathered, the sequence of actions is executed and the result is returned to the CREEK agent.

Application agents are autonomous agents that are capable of executing one or more actions. An action may range from the mundane to services requiring complicated reasoning. Each application agent corresponds to an artefact in the specific domain, which offers one or more services.

A more thorough description of the overall architecture can be found in [14,19], the overall agency is described in detail in [20], and the decomposition of tasks and execution of actions are described in [21].
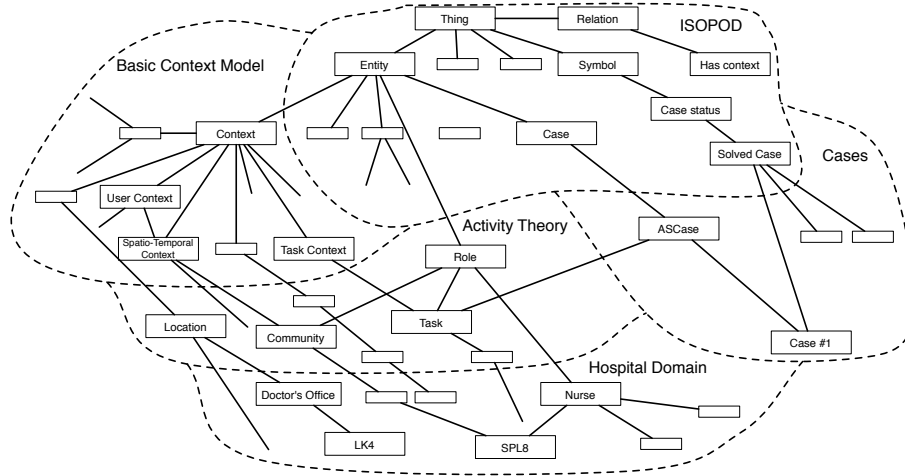
Fig. 3. The Integrated Knowledge Model

3   Context as Lenses

Context is used in two distinct ways, as illustrated in Figure 2. It is initially used as a focus lens on the part of the world that can be perceived. In this sense, context is the part of the world that is available to the awareness part for identifying the situation. The part of the world depicted as Perceived in the *Perception* part of Figure 2, and as Situations Context in the *Awareness* part, are the parameters CREEK uses in the retrieval part of the CBR cycle. These parameters can be the location where the situation is occurring, the people present, and the time of day. A more elaborate example is given in Section 7.

The other use of context is for solving the particular problem associated with a situation. Once CREEK has identified the situation, the corresponding goal is also identified. For example, when a *pre-ward round* situation has been identified, it is also known that the goal of the situation is to *evaluate treatment* of the patient. Along with the goal, is a sequence of tasks that will achieve the goal. The different artefacts or persons that can contribute by executing the actions that correspond to the tasks in the sequence, may differ depending on the context. Thus, the context, as depicted in the *Sensitivity* part of Figure 2, is viewed as a focus lens on the world given by the goal in question. This represent a further constraint on the information available. If, for instance, the initial task in a sequence is to acquire the name of the patient, and this information can be obtained from several sources, it is important to know what information sources are available in the given situation. In this example the name of the patient might be obtained from the patient chart, the nurse, or the electronic patient record; however, if context contains only the patient chart and the nurse, then the electronic patient record can't appear in the sequence of actions.

4 Knowledge Model

The CREEK system follows a knowledge intensive approach to Case-based Reasoning. There is a strong coupling between the case specific knowledge and the general domain knowledge, as the cases share the same multi-relational semantic network as the domain knowledge [22]. This integration of specific and general knowledge requires a strong focus on knowledge acquisition and modelling [23].

Our knowledge model has been constructed partly top-down and partly bottom-up. As Figure 3 illustrates, all knowledge types are integrated into a single semantic network. Each node is a concept described in a frame structure. The three parts generic to any CREEK application are the top-level ontology (called ISOPOD), the domain-specific model of general and factual knowledge, and the case base. The two subparts developed for the ambient intelligence system are the Basic Context Model and the Activity Theory model.

As one of the most important aspects of situations are the activities occurring, Activity Theory was used to capture these aspects of the world and integrate them into the knowledge model. This part is concerned with notions such as roles, artefacts, and communities, and their relationships. For a thorough discussion on the use of Activity Theory for context modelling see [24,25].

This Basic Context Model is structured around a taxonomy inherited from the context-aware pervasive computing tradition [26]. The model is divided into five main aspects describing a situation: *i*) Personal Context, describing information about the user of the system; *ii*) Task Context, which is concerned with describing the task performed by the user, and other entities to achieve the goal; *iii*) Social Context, detailing the roles of the user; *iv*) Spatio-Temporal Context, which holds information on location, time and the community shared by the user; *v*) Environmental Context, describing the entities present, such as other persons and artefacts.

The model was populated through observations done at the St. Olav Hospital in Trondheim through the summer of 2005. Some of the data gathered were used in a bottom-up approach to enrich the model. The enrichment included: the different locations at the wards; the roles that employees, patients, and visitors assumes; the classes of persons encountered in the wards; artefacts and services they offer and consume.

5 Case Model

A case has two main parts: the *findings* of the situation, which consists of the perceived context; and the *solution*, which consists of the goal to be achieved and the corresponding task to be accomplished. Note that, in this application, to interpret a situation involves identifying its goal, i.e. to determine what is to be done in the situation. A situation is a meeting between health personnel, and these meetings always have a purpose, or a goal, which may be to agree on a diagnosis for a patient, to decide whether a particular examination should be made, or evaluate what treatment to give. An example of a situation instantiated, but not yet classified, in CREEK, is shown in Figure 4.

```
(UnsolvedCase001 (
  has-case-status: Unsolved
  has-context: Context:UserContextSituation001 (
    context-of: Person:User:AL7
    has-part: SpatioTemporalContext:SpatioTemporalContextSituation001 (
      has-part: Location:DoctorsOffice:LK4
      has-part: Time:0915
      has-part: Community:Situation001Community (
        has-part: Role:PatientResponsibleRole
        has-part: Role:GroupLeaderRole))
    has-part: SocialContext:SocialContextSituation001 (
      has-part: Role:PatientResponsibleRole)
    has-part: EnvironmentalContext:EnvironmetalContextSituation001 (
        has-part: Person:HealthWorker:Nurse:SPL8
        is-present: Artefact:PatientList
        is-present: Artefact:PatientChart
        is-present: Artefact:PAS)
  has-goal:
  has-task: ))
```

Fig. 4. Unknown situation

This case describes a situation where several things are known. All of the findings of the case are described as the *has-context* part. It describes a context for *AL7*, who is the user of this system. It is known that the situation is occurring in *LK4*. If we look at the knowledge model, we will discover that LK4 is a specific *Doctor's office*, which again is a type of location. The situation is occurring at 09:10, and the *community* consists of two roles, the *PatientResponsibleRole* and the *GroupLeaderRole*, which are played by AL7 (the user) and SLP8 (a nurse).

The example above is taken from our data set (see Section 6), thus all possible parameters are not included. A case is a specific view on the knowledge contained in the system. This is a key feature of the system. The Awareness layer (see Figure 1) constrains the part of the knowledge base that the Sensitivity layer should rely on to solve the problem in particular situations.

Once a situation has been classified, or in other words, a matching case has been retrieved, the solution is acquired from the matching case. This solution contains the goal of the situation and the main task which achieves this goal.

The exact sequences of tasks used to accomplish the goals are not stored directly in the cases. They are stored and maintained in the decomposer agent. Since the nature of a goal does not vary significantly within situations of the same type, it was not deemed necessary to adapt the task sequence itself as part of the reasoning cycle. Rather, we assume that instead of the tasks varying, the artefacts and communities required to execute the actions vary across situations of the same type. Thus, as a rule, the task sequence does not need to be redefined for each situation, rather a generic sequence can be populated with willing and able agents for each situation (for more on the execution of plans see: [21]). Along with the goal and the main task, the artefacts and communities available can now be transmitted as context to the decomposer agent, i.e. the second use of context, as described in Section 3.

6 Data Set

The applicability of the overall architecture has been demonstrated in [15], where a test of a stimuli-response type behaviour was conducted. For various reasons this test did not include the full case-based reasoning cycle. One of the most important reasons was a lack of suitable situation data. The lack of data has now been remedied, as a large amount of data on situations has been gathered.

Table 1. Context parameter describing situations

Location	The room where the situation occurred
User	The user of the system
Role	The role of the user
Present	Other persons present
Role	The role of each of the persons present
Patient ID	The ID of the patient in question
Time	The time of day

The data was collected through a period of one month at the St. Olav Hospital in Trondheim. A medical student followed several employees and recorded the situations that occurred throughout the days. Two wards were studied: the cardiology and gastroenterology ward. The data set contains 197 situations for cardiology and 163 for gastroenterology. Approximately two-thirds of the situations from cardiology concerning the consultant physician (OL9) were transferred into the system, and classified, manually. This set of situations gives the system a set of initial cases to reason about. The last third are used to test if the system can classify the situations, identify the sequence of actions, and execute them.

The data gathered was structured into two main types: the type that is primarily related to comprehending the situation (Table 1), and the type primarily related to achieving the goal and constructing the task structure (Table 2).

Table 2. Context parameters for problem-solving

Source	Information sources and targets
I/O	The direction of the information flow
Information	Type of information

The data describing the context of situations (Table 1) include some information which could easily be sensed though available hardware. These are the location where the situation is occurring; the user whose perspective we are adopting; other persons present; and the time of the situation occurring. It is debatable if the role of the persons present, and the ID of the patient in question

(whom is not necessarily present), are readily available for automatic sensing. However, we have regarded this issue as problems of engineering, and not critical for testing the applicability of our method.

The context parameters describing the situation are modelled as part of the case-base, as exemplified in Figure 4. These data are primarily used during the *retrieve* phase of the cycle. However, being part of the general knowledge model, they are also available to the decomposer agent for problem-solving.

The problem-solving data (Table 2) is used for two purposes. They are first used to populate the knowledge model with information on the artefacts (sources) present and the services (information) they offer and consumes. Secondly, they are used in the decomposer agent for constructing action sequences. Initially they are constructed and generalised manually.

Currently, the data for the consultant physician OL9 at the cardiology ward has been incorporated into the system. The 197 situations, including all the physicians, are distributed as described in Table 3

Table 3. Distribution of observed data for cardiology

Situation	AL7	AL9	AL14	OL9	Sum
Pre-pre-ward-round	5				5
Pre-ward-round	7	22	11	26	66
Ward-round	7	21	11	26	65
Examination		8	2	9	19
Post-work		8	9	13	30
Pre-discharge			2	4	6
Heart meeting		1		1	2
Discharge meeting				4	4

Eight different types of situations have been identified in the data set. Four different physicians were observed, where three were assistant physicians (AL7, AL9, AL14) and one was a consultant physician (OL9). Beside these, several nurses, patients, and relatives are present in different situations.

7 Example

To clarify how this system is used to identify situations and solve the problem associated with a situation, this example will demonstrate the chain of events occurring when an unknown situation is presented to the system. First, we exemplify a stored case, then we show how the case is utilized after having been retrieved by CREEK's two-step similarity assessment method [22].

As described in Section 6, a sub-set of the observed situations are used to populate the knowledge model with initial cases. In this example a case describing a pre-ward round situation is used. Figure 5 describes its findings and solution.

```
(ASSolvedCase01 (
  has-case-status: Solved
  has-context: Context:UserContextAL7Previsit001 (
    context-of: Person:User:AL7
    has-part: SpatioTemporalContext:SpatioTemporalContextAL7Previsit001 (
      has-part: Location:DoctorsOffice:LK4
      has-part: Time:0910
      has-part: Community:PrevisitCommunityAL7001 (
        has-part: Role:PatientResponsibleRole
        has-part: Role:GroupLeaderRole))
    has-part: SocialContext:SocialContextAL7Previsit001 (
      has-part: Role:PatientResponsibleRole)
    has-part: EnvironmentalContext:EnvironmentalContextPrevisit001 (
        has-part: Person:HealthWorker:Nurse:SPL8
        is-present:Artefact:PatientList
        is-present:Artefact:PatientChart
        is-present:Artefact:ElectronicPatientRecord)
  has-goal: EvaluateTreatment
  has-task: EvaluateTreatmentTask))
```

Fig. 5. Pre-ward-round initial case

A pre-ward round is a type of meeting that occurs every morning between the physician on duty and the nurse in charge. In this meeting the patients on the ward are discussed and the treatment is evaluated to define the further treatment for the day in question. These types of meetings typically include all the patients in the ward. However, we have chosen to divide the meeting into distinct situations, each containing the evaluation of one patient. In other words, the system perceives the evaluation of each patient as one situation, so each meeting may consist of several situations.

Along with these findings, the goal of the situation is stored in the case. In the example of the pre-ward round, the goal of *evaluate treatment* has been manually identified in the observed data. From the examination of the observed data, we have also identified the typical sequence of tasks corresponding to the evaluation of the patient's treatment:

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

Initially, the identification of the patient in question is acquired. Based on this identification, any changes in the patient's condition is mapped out. Now, the medication scheme and any new test results are examined. Finally, any changes in the treatment, such as changes in medication or scheduling of new tests, are noted. Once the case based has been populated with a set of cases, a new and unknown situation – shown in Figure 4 – is presented. The case in Figure 5 is returned as the best match.

Comparing these two situations, it is evident that they have a strong resemblance. The only significant difference is the time when the situations occur.

Hence, it is no big surprise that CREEK infers that these two situations are highly similar; actually it concludes that the matching strength is 88% (see Figure 6, in which the explanation structure supporting the match is also shown).
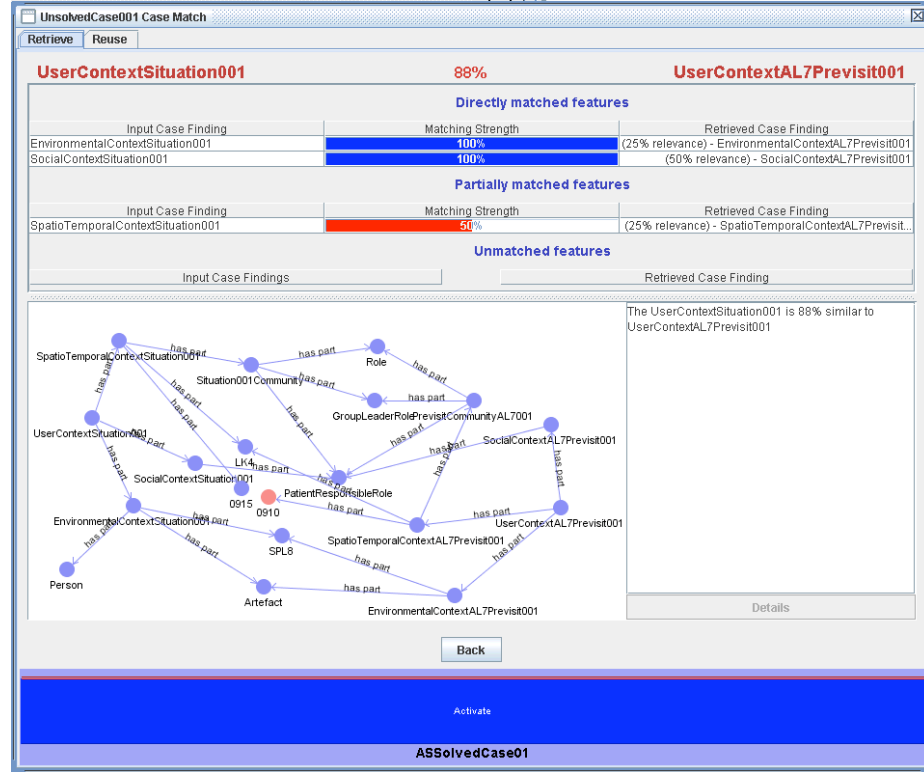

Fig. 6. Matching Cases

CREEK has now determined that the most likely candidate for identifying the situation is a pre-ward round. Knowing this, CREEK can also tell the sensitivity part of the system that the goal of this situation is to *evaluate treatment*. However, as noted earlier, even though the sequence of tasks are the same, the exact actions required can differ from the original situation. So to achieve the best possible execution of the sequence of actions, the context in which the tasks occur is transmitted along with the goal. In this case, this context is the observed parameters, as well as the artefacts present in this situation. In the case of this newly classified pre-ward round situation, these artefacts are: the list of patients, able to supply the name of the patient; the patient chart, able to supply the medication plan, the PAS system (a patient administration system), and of course the nurse.

This context guides the task decomposition process in the sensitivity part. In this example there is no reason to enquire the electronic patient record for any results of a blood sample, as only the patient chart and the nurse are present. The decomposer can now select the plan for evaluating the patient's treatment, and draft the application agents that are able and willing to execute the actions corresponding to the tasks in the plan.

In this example the task was to initially acquire the name of the patient in question. Looking at the artefacts present, as described in the knowledge model, the only one capable of supplying this information is the *patient list*, so the corresponding agent is drafted. Following the identification of the patient, any changes in the patient's condition must be identified. Only two entities are capable of supplying this type of information, namely the *nurse* or the *patient chart*. The next step is to examine the medication plan. Here, only the *patient chart* is capable of supplying the necessary information. Once the medication has been examined, the next step is to acquire any new test results. Looking at the context there are two artefacts capable of delivering this information: the *patient chart* and the *PAS* system. Finally, any changes to the patient's treatment plan must be noted, and both the *nurse* and the *PAS* system are capable of this.

The left side of Figure 7 depicts the observed chain of actions in the original known situation. The right side depicts the possible artefacts that can supply the information required in this context. When we compare the two different instances of a pre-ward round situation, the main difference is that we do not know who will actually supply any changes in the patient's condition (point 2 in Fig. 7), and the fact that any new results can only be acquired from the patient chart, and not the electronic patient record (point 4 in Fig. 7).

1. Acquire name of patient
   Patient list
2. Acquire changes in patient's conditions since yesterday
   Nurse
3. Examine, and possibly change, medication scheme
   Patient chart
4. Acquire any new results from tests
   Patient chart, electronic patient record
5. Note changes in treatment
   Nurse

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

Fig. 7. Two sequences of actions

This combination of the goal received from the CREEK agent, along with the context, allows the decomposer agent to generate the sequence of actions depicted on the right side of Figure 7, as well as executing it using the agents

drafted. To verify that the identification of the situations, as well as the generated tasks, are sensible we can look at the observed data, to figure out what really happened. In the observed data the situation that we just classified had the following sequence of actions:

1. The *patient list* supplied the patient's name
2. The *nurse* informs about the changes to the patient's condition
3. The *patient chart* supplied the medication scheme, which was changed
4. The *nurse* informs about the result of a blood sample
5. The *PAS* system was informed about result of this pre-ward round

When we compare this real world sequence with the possible sequence proposed by the decomposer, we can observe that it easily encompasses the one that actually happened.

8   Conclusion and Future Work

This paper outlines how a knowledge intensive Case-Based Reasoning system can be constructed to facilitate ambient intelligence. It has been shown how a combination of generic concepts for reasoning and modelling, a context taxonomy, and elements from Activity Theory, can influence the way a knowledge model for situation awareness can be implemented. A case model has exemplified how cases should be structured with context as the problem defining part, and the general goal and task as the solution to the implicit problem in situations.

There are clear indications that ambient intelligent systems can benefit from the socio-technical analysis made by applying Activity Theory. This is beneficial when defining the knowledge model, as well as when constructing guidelines for observations.

It has been demonstrated how all the above parts can be combined into a coherent architecture, which allows for the perception of the environment, reasoning about context to identify situations, and problem-solving based on this understanding.

Based on the data we collected for the cardiology ward, we are currently populating the case based with more cases; as well as constructing the generic task sequences corresponding to the eight different situations discovered. The next step is to do an initial verification of the system's integrity, and test CREEK's ability to correctly identify new cases; before executing a full fledge simulation of the system.

Some issues have intentionally been left out of this description. Most importantly the question of how CREEK may learn when it is run on-line. Several approaches have been discussed, such as simply asking the user, or trying to perceive the user's behaviour based on the recommendations given by the system. Each has its benefits and drawbacks. Asking the user directly is likely to be the technical easiest approach, however, this can very easily be quite intrusive for the user. Attempting to perceive the behaviour by looking at the situations

perceived is most likely the most complicated, but also the least intrusive. However, recent studies on conversational CBR in our group show promising results for optimising user querying strategies [27] by combining observation and conversation.

9 Acknowledgements

References

1. Weiser, M.: The computer for the 21st century. Scientific American (1991) 94–104
2. Satyanarayanan, M.: Pervasive computing: Vision and challenges. IEEE Personal Communications **8** (2001) 10–17
3. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for Ambient Intelligence in 2010. Technical report, IST Advisory Group (2001)
4. Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to Context than Location. Computers & Graphics Journal **23** (1999) 893–902
5. Liu, H., Maes, P.: What would they think? In: Proceedings of the 2004 International Conference on Intelligent User Interfaces, ACM Press (2004) 38–45
6. Patrick, B.: Task-realization models in contextual graphs. In Dey, A.K., Kokinov, B., Leake, D., Turner, R., eds.: Modeling and Using Context: 5thInternational and Interdisciplinary Conference CONTEXT 2005. Volume 3554 of Lecture Notes in Computer Science., Spriger Verlag (2005) 55–68
7. Öztürk, P., Aamodt, A.: A context model for knowledge-intensive case-based reasoning. International Journal of Human Computer Studies **48** (1998) 331–355
8. 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
9. 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
10. 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
11. Vygotsky, L.S.: Mind in Society. Harvard University Press, Cambridge, MA (1978)
12. Plaza, E., Arcos, J.L., Martín, F.J.: Cooperative case-based reasoning. In Weiss, G., ed.: Lecture Notes in Artificial Intelligence. Volume 1221 of Lecture Notes in Computer Science. Springer Verlag (1997) 180–201
13. Díaz-Agudo, B., González-Calero, P.A.: An architecture for knowledge intensive cbr systems. In Blanzieri, E., Portinale, L., eds.: Advances in Case-Based Reasoning: 5th European Workshop, EWCBR 2000. Volume 1898 of Lecture Notes in Computer Science., Springer Verlag (2000) 37–48

14. Myrhaug, H.I., Whitehead, N., Göker, A., Fægri, T.E., Lech, T.C.: AmbieSense – A System and Reference Architecture for Personalised Context-Sensitive Information Services for Mobile Users. In Markopoulos, P., Eggen, B., Aarts, E., Crowley, J.L., eds.: Ambient Intelligence: Second European Symposium on Ambient Intelligence, EUSAI 2004. Volume 3295 of Lecture Notes in Computer Science., Springer Verlag (2004) 327–338

15. 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

16. Endsley, M.R., Bolté, B., Jones, D.G.: Designing for Situation Awareness: An Approach to User-Centered Design. Taylor & Francis (2003)

17. Aamodt, A.: A knowledge-intensive, integrated approach to problem solving and sustained learning. PhD thesis, University of Trondheim, Norwegian Institute of Technology, Department of Computer Science (1991) University Microfilms PUB 92-08460.

18. Fensel, D., Motta, E., Benjamins, V.R., Crubezy, M., Decker, S., Gaspari, M., Groenboom, R., Grosso, W., van Harmelen, F., Musen, M., Plaza, E., Schreiber, G., Studer, R., Wielinga, B.: The unified problem-solving method development language upml. Knowledge and Information Systems **5** (2003)

19. Kofod-Petersen, A., Mikalsen, M.: An Architecture Supporting implementation of Context-Aware Services. In Floréen, P., Lindén, G., Niklander, T., Raatikainen, K., eds.: Workshop on Context Awareness for Proactive Systems (CAPS 2005), Helsinki, Finland, HIIT Publications (2005) 31–42

20. Lech, T.C., Wienhofen, L.W.M.: AmbieAgents: A Scalable Infrastructure for Mobile and Context-Aware Information Services. In: AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM Press (2005) 625–631

21. 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

22. 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

23. Aamodt, A.: Modeling the knowledge contents of CBR systems. In: Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning, Vancouver. Naval Research Laboratory Technical Note AIC-01-003 (2001) 32–37

24. Kofod-Petersen, A., Cassens, J.: Activity Theory and Context-Awareness. In Schulza, S., Leake, D.B., Roth-Berghofer, T.R., eds.: Proceedings of the IJCAI-05 Workshop on Modeling and Retrieval of Context (MRC 2005). Volume 146., CEUR Workshop Proceedings (2005) 1–12

25. 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)

26. Göker, A., Myrhaug, H.I.: User context and personalisation. In: Workshop proceedings for the 6th European Conference on Case Based Reasoning. (2002)

27. 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)