Reasoning From Contexts

Andreas Zimmermann, Andreas Lorenz, Marcus Specht
Fraunhofer Institut for Applied Information Technology
Schloß Birlinghoven
53754 Sankt Augustin, Germany
{Andreas.Zimmermann, Andreas.Lorenz, Marcus.Specht}@fit.fraunhofer.de

Abstract

The paper describes a new approach of using the Case-Based Reasoning (CBR) methodology in context-aware systems. We elaborate how this technique can be applied to generate recommendations based on the contexts of users respectively objects especially in a mobile scenario. By combining CBR methodology and context awareness a new and powerful way of modelling and reasoning from contexts emerges. Using CBR cases to enclose the context will then enhance the possibilities to compare contexts, determine certain values of context-similarities, and reflect this information in the process of generating recommendations.

1 Introduction

Context as a mean for adaptation of information selection and presentation has been described in a variety of ways and approaches [7, 8, 9]. Nevertheless the underlying problem of identifying similarities and differences between different constellations of context parameters has not been discussed intensively in the literature. From our point of view identifying important context parameters to describe a user interaction is an essential issue when designing context aware information services.

The following sections describe the application of a well known methodology to a mobile scenario. In our approach, snapshots of the domain and Case-Based Reasoning are used for representation and reasoning about contexts. These snapshots discretize a highly dynamic domain and describe the different contexts or situations in which a user interacts with an information system. They form the basis for identifying important features of situations individually for each user. In addition, by using this methodology we try to find similar situations in the history of a user, or a group of users, and to give appropriate context-aware recommendations to mobile users.

Furthermore another basic problem of personalized information systems is modelling of user interests in environments where the focus of users is highly dynamic. With the described approach we try to model user's changing focus and his/her focus of interest. Therefore we have chosen a case-based reasoning approach where situations are modelled as cases and different aspects of a case can be weighted and taken into account differently for computing recommendations and user interest model. In our understanding important features of a rapidly changing user interest model can be represented adequately with a weighting of context parameters of the current user situation.

Case-based Reasoning may be suitable for problem areas in which the knowledge of how a solution is created is poorly understood. Case-based reasoning has been applied to a variety of application domains such as: design support [2], software re-use [3], industry applications [4], electronic commerce [5], and help-desk support systems [6]. There are already a number of CBR systems in commercial use.

Aamodt and Plaza illustrate the main ideas of Case-Based Reasoning in [1]. The central aspect of Case-Based Reasoning is problem solving based on the similarity to past experiences. These experiences are stored as cases in the case base and consist of a problem description and a solution description. As pointed out in [10], user contexts are difficult to map to a problem-solution pair. In the following we describe one approach of how this can be managed.

2 Context modelling

Several approaches have defined context models and described different aspects of context taken into account for context-aware systems. For example, Schilit et al. have mentioned [8]: where you are, who you are, and what resources are nearby. Dey and Abowd [9] discuss several approaches for taking into account the computing environment, the user environment, and the physical environment and distinguish primary and secondary context types. Primary context types describe the situation of an entity and are used as indices for retrieving second level types of contextual information.

In this work we base our context modelling approach on the four dimensions of a context which Gross and Specht have considered in [7]:

- *Identity*: The identity of a person gives access to the second level of contextual information. In some context-aware applications highly sophisticated user models hold and infer information about the user's interests, preferences, knowledge and detailed activity logs of physical space movements and electronic artefact manipulations. The group of people that shares a context can also define the identity of a context.
- *Location*: We consider location as a parameter that can be specified in electronic and physical space. An artefact can have a physical position or an electronic location described by URIs or URLs. Location-based services as one type of context aware applications can be based on a mapping between the physical presence of an artefact and the presentation of the corresponding electronic artefact [8].
- *Time*: Time is an important dimension for describing a context. Beside the specification of time in CET format categorical scales as an overlay for the time dimension are mostly used in context-aware applications (e.g., working hours vs. weekend). For nomadic information systems a process-oriented approach can be time dependent (similar to a workflow).
- *Environment or Activity*: The environment describes the artefacts and the physical location of the current situation. In several projects approaches for modelling the artefacts and building taxonomies or ontology about their interrelations are used for selecting and presenting information to a user.

2.1 Concept

In this paper we exemplify our ideas with a context-aware museums guide. Consider a person walking through a museum and having some special interests in arts. The context-awareness of the system should result in the ability to find visitors that share the same interests (e.g. paintings of August Macke), or art works that satisfy the users interests. Both other visitors and art works share their contexts with the current user and are therefore recommended to the user by the system.

In our approach we apply Case-Based Reasoning techniques in order to find similar (i.e. shared) contexts in a mobile scenario and to submit recommendations based on this similarity. Thus, an appropriate case model needs to be built to adequately represent the above mentioned parts of a context.

We choose a differentiated view on the domain: We assume, that every object located in the physical or in the electronic space [7] has its own context. There is no explicit distinction between users and other objects within the domain. Some objects like exhibits or restaurants in a museum hold a *static context*, which does not change very often. Other objects like people moving around own a highly *dynamic context*, which changes frequently.

The basic idea now is to model the different contexts of the domain objects in a way that they can be compared to each other by a CBR system in order to retrieve similar contexts. If the similarity of two object contexts transcends a certain value, the information held in the context of one object serves as the recommendation for the other (see Section 4.2). But before a recommendation can be announced, static and dynamic contexts are to be modelled as cases of a CBR system.

2.2 Modelling Cases

Because the case-based reasoning cycle is based on the determination of similarities between cases, objects of the application domain are to be mapped to cases in a case-based manner. In addition, a vocabulary is needed to describe contexts in detail and to move them to a general, more abstract level, on which they can be compared. The most common modelling technique in CBR systems is an object-oriented hierarchy of attribute-value pairs, which is a powerful "language" for expressing contexts. The context of a domain object is packed into a n-dimensional vector of values, each representing information about an attribute of the object.

Following a straight forward approach we map the problem-solution pairs to context-recommendation pairs as shown in Figure 1. The problem description complies with the description of the context and the problem solution is an object identifier. This identifier is associated with its specific context description, which consists of a static and a dynamic part and serves as recommendation.

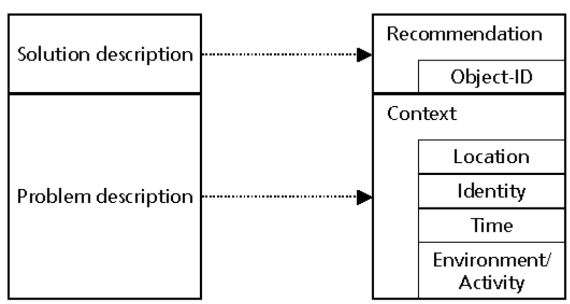

Figure 1: Case model

In our example scenario the contexts of visitors of a museum and the contexts of the exhibits have to be modelled as cases. For a person walking through a museum such attributes characterize his/her age, address and interest in artworks (cf. Figure 2). This static part of the context description is mainly represented by the identity dimension as described above. The dynamic part of the context may be some attributes describing the visitor's spatial position in the exhibition hall and the time of the day, because they alter their values frequently. These attributes correspond to the above mentioned location dimension.

Both sets of attributes are combined within one context description and are to be transformed into a case of a CBR system. Dynamic attributes raise a problem, because CBR systems cannot handle continuous domains. They do mainly work on discreet point-cases in a n-dimensional space. The next section presents our concept to overcome this problems.

2.3 The Snapshot-Approach

As mentioned before, CBR systems cannot handle continuous domains. Every sensor scanning the environment is connected to one attribute of the context description. One sensor may deliver an update value for its associated attribute on a time slice or event basis (e.g. a position tracking system may measure the position of an object and deliver an update value every 10 ms).

A CBR system mainly processes point-cases. Hence, we need a technique that performs a discretisation of a continuous domain. For this purposes we propose taking snapshots of the domain in discrete time intervals. Like a camera freezes the current environment and captures it on a picture, a snapshot freezes the current values of all sensors and assigns them to their corresponding attributes of the context description. Obviously, the granularity of the delivered information depends on the frame rate of the snapshots. Thus, in combination with all static attributes of a context, this n-dimensional vector of static attributes is transferred into one single case. This case can be stored in the case base of a CBR system and is available for further processing.

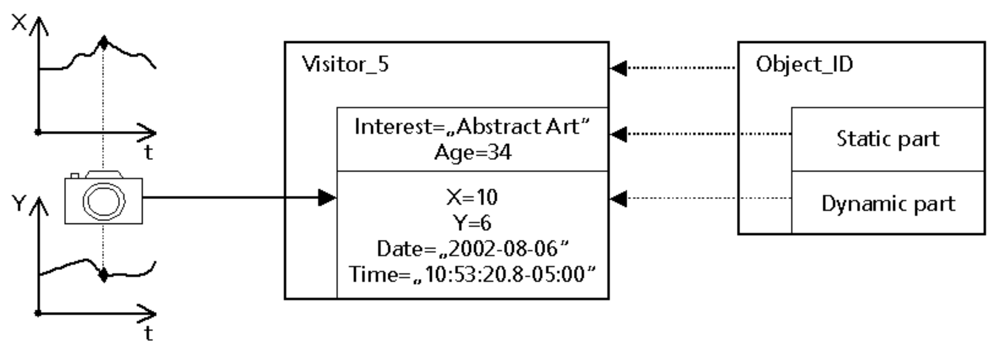

Figure 2: Snapshot of a person walking through a museum

Figure 2 illustrates an exemplary context-snapshot of a person walking through a museum. It captures his/her position (x,y) in the spatial environment and the point in time indicating when the snapshot was taken. This snapshot is enriched by static information, in this example the person's interest in artworks and his/her age, and finally stored as a case in the case base. If a snapshot were taken every five seconds, the moving person would leave a trace of cases in the case base.

3 Generating Recommendations

In the preceding section we have described a mobile context modelling approach using the CBR system. This section now introduces how the system generates recommendations and how domain objects with a more or less static context are to be treated.

3.1 Procedure

In Section 2.2 we discussed the representation of a context as a n-dimensional vector stored as a case in the case base of a CBR system. In a n-dimensional space, such a vector refers to a certain point. Based on the distance measure used for the similarity assessment of the CBR system, around each point case a sphere emerges, whose appearance depends on the weights assigned for the several attributes. In the distance measure we use, all attributes have the same weights and so we are working with symmetric spheres only. With different weights, the spheres may appear in more abstract shapes.

If the context of a certain object changes, for example because its position has changed, the CBR system works like a collision detection component reporting when two, or more, spheres overlap. This indicates that the contexts of objects are similar to each other – technically, the similarity measure has exceeded a threshold value.

If two contexts are similar, we define a recommendation to one domain object as a link to the other object that is associated with its specific context. As described above, we do not distinguish between users and objects. In our approach we differentiate between active objects and objects, which do not take action themselves. Both kinds of objects may be destination for a recommendation. In most cases, the destination objects for recommendations are users, because they are mobile and for this reason active. But behind the recommended object identifier some static object (such as an exhibit, a restaurant, or a gas station) may also be hidden. For example, a recommendation to a restaurant could be a message like this: “Promote your Italian food; some tourists are approaching!”

Example:

> In our exemplary application users are visiting an exhibition, either in the physical or in any electronic space. Our system should provide valuable information to the visitor about certain exhibits or visitors sharing his/her context. If the spheres of two context descriptions overlap in the n-dimensional space, the CBR system detects this collision. The system recommends the exhibit’s or visitor’s identifier to the observed user. In our case the recommendation message consists of the identifier and the values of all attributes associated with this identifier. After that a flag indicates that the two objects were recommended to each other.
>
> Note that neither the current user nor the recommended object is forced to be real world objects of the physical space; both objects can also be virtual objects of the electronic space (cf. Section 2.1; [7]). A recommendation to a user in the electronic space may also be a link to visit the electronic artefact, or even an invitation to come to the museum and have a look on the physical one. In the same way, users in the electronic space and real world users also can come together. If they share contexts, a medium of communication is recommended. If both the users are physically visiting the museum, they can just talk to each other, or otherwise, they may take e-mail addresses or phone numbers from the particular context description.

It depends on the implementation of different views, how the context of a recommended object can be interpreted to really express or formulate a recommendation. In order to realize a collaborative view the system may send an email to all users sharing similar contexts. This email would contain information on why a contact should be established at each case. A recommendation view would implement the transmission of a message to the mobile device of the user explaining why the restaurant a few steps ahead is so special.

An additional view that is not taken into account in our approach has to be mentioned: the experiences the system stored about the user, i.e. the cases in the case base that concern this specific user. These experiences can be viewed as a model of the user and used to refine certain recommendations. In addition, by clustering the set of cases associated with one user points of interest can be uncovered.

3.2 Static Objects

In Section 2.1 we have mentioned that some objects like exhibits in museums do not change their contexts often. Furthermore these objects are not active, i.e. they own a very static context, and thus the system cannot draw any conclusions from their behaviour. Users of the system can give some information about their own context to the system by themselves. In addition, by observing users the system can draw conclusions from the user's behaviour. In turn, these conclusions may influence the representation of the users' interests, knowledge, and probably needs.

In contrast to the user context, the context of static objects needs to be defined beforehand. Therefore an administrator of the case base will be necessary to define the context for static objects, formulate some recommendations for different views, and finally create cases. Thus, the initial case base consists of a number of fixed cases.

4 Conclusion and Future Work

In this paper we have introduced a method for reasoning from contexts of mobile users respectively objects. The abstract idea is to create a n-dimensional space (including the time and position dimension), in which points represent a certain context or situation. In a mobile scenario some points move in space (active objects) and some do not (static objects). An underlying Case-Based Reasoning methodology determines, when two or more points get close to each other. In our approach the well known Case-Based Reasoning methodology is used in a standardized way based on equal weights for all attributes. As shown in an exemplary environment, i.e. visitors walking through a exhibition hall with a mobile device, this procedure will be useful in the field of context-aware systems.

Based on this approach, a multifaceted field of possibilities for future work emerges. First, the way recommendations are generated or formulated should be improved. The information represented by the context description should be exploited by a subsequent application and used to build meaningful recommendations. Furthermore, this application should implement different views on the context.

Second, the knowledge containers of the CBR system should be augmented with learning techniques. On the one hand, the attribute weights for the similarity measure should be learnt by some machine learning algorithm and adapted to the users behaviour. In addition, different vectors of weights can be used to model different user views on the domain objects. On the other hand, the case base has to be improved. A known technique for forgetting cases may be exploited or sets of very similar cases can be replaced by only one generalized case, which is an open research field. From the learning of individual weights of context parameters we expect a significant improvement of the recommendations given to users with fast changing interest focus.

Another interesting research topic is the expansion of the shared context metaphor: How should users that share their contexts for a longer time period be treated (e.g. a couple travelling by car)? Should their contexts be merged into one context of a new single object and how could this be achieved? How do recommendations to this newly emerged object look like? This scenario explicitly makes use of the idea of treating everything of the domain as an object.

5 References

[1] A. Aamodt, E. Plaza. *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*. AI Communications, vol. 7 no. 1, pp 39-59, 1994

[2] M. Pearce, A.K. Goel, J.L. Kolodner, C. Simring, L. Sentosa, R. Billington. *Case-Based Design Support*, IEEE, pp. 14-20, October 1992

[3] G. Fouqué, S. Matwin. *Compositional Software Reuse with Case-Based Reasoning*. Conference on Artificial Intelligence Applications 1993, IEEE, Florida, 1993

[4] R. Bergmann, S. Breen, M. Göker, M. Manago, S. Wess. *Developing Industrial Case-Based Reasoning Applications. The INRECA-Methodology*. LNAI 1612. Springer Verlag. 1999

[5] S. Schmitt, R. Bergmann. *Applying Case-Based Reasoning Technology for product selection and Customisation in Electronic Commerce Environments*. In Proc. of the 12th Bled Electronic Commerce Conference, Bled (Slovenia), June 1999

[6] M. Göker, Th. Roth-Berghofer. *Development and Utilization of the Case-Based Help-Desk Support System HOMER*. In Proc. of the 3rd International Conference on Case-Based Reasoning, ICCBR '99, 1999

[7] T. Gross, M. Specht. *Awareness in Context-Aware Information Systems*. In Mensch & Computer – 1. Fachübergreifende Konferenz (March 5-8, Bad Honnef, Germany). H. Oberquelle, R. Oppermann and J. Krause, Eds. Teubner, 2001. pp. 173-182.

[8] B.N. Shilit, N.I. Adams, R. Want. *Context-Aware Computing Applications*. In Proceedings of the Workshop on Mobile Computing Systems and Applications (IEEE Computer Society, Santa Cruz, CA, 1994. pp. 85-90.

[9] A.K. Dey, G.D. Abowd. T*owards a Better Understanding of Context and Context-Awareness*. GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf, 1999. (Accessed 12/12/2000)

[10] A. Göker, H. Myrhaug. *User Context and Personalization*. In Proc. of the 6th European Conference on Case-Based Reasoning, Aberdeen (Scotland), September 2002