Building a Personal Memory for Situated User Support

Michael Schneider, Mathias Bauer, Alexander Kröner

German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbruecken, Germany
{michael.schneider, mathias.bauer, alexander.kroener}@dfki.de

ABSTRACT

Keeping a history of the user's interaction with the environment is of use for many reasons. However, collecting, structuring, accessing, and reviewing such potentially large amounts of information is not trivial. In this paper we present our ideas for a memory model for pervasive computing applications addressing these questions. The proposed architecture allows applications to deliver ad-hoc support taking into account the user's history and general attitudes as well as providing a personal diary to review events and retrieve memories. We also include a brief discussion of a novel user interface, which allows the user to bind services to general contexts based on her previous experiences.

Keywords

Context histories, user modeling, adaptive user support

INTRODUCTION

The diffusion of sensor technology from dedicated devices into our everyday environment offers a potentially omnipresent, rich source of information that might be used by pervasive computing applications in multiple ways. An example of such an application is an artificial memory extending the user's perception. With such a memory, on the one hand context dependant support can be provided to the user by considering experiences previously made in similar situations. On the other hand, such an artificial memory could complement the user's natural memory and could be used to retrieve forgotten or unnoticed information at a later point.

On the way to this ambitious goal three main research questions arise:

1. How is useful information identified and acquired?
2. How is stored information structured/organized?
3. How is memory content retrieved and reviewed?

All the work described in this paper is conducted within the project SPECTER. Goal of the project is to build a personal ubiquitous assistant, which keeps an artificial memory of the user's experiences in order to deliver ad-hoc and subsequent context dependant support. As such, SPECTER has to deal with all of the above questions. However, in this paper, we focus on the second question and present some concepts related to the third question. For reasons of completeness we will include a rather short and practical discussion of the first question.

Since the project SPECTER is still running, not all of the ideas presented in this paper have been fully implemented yet. Therefore, we will give implementation details where possible and discuss our theoretical ideas otherwise.

The rest of this paper is structured as follows. After a description of the demo scenario used for the examples in this paper, we present a practical approach to the information acquisition problem. We continue with the description of the memory model used in SPECTER to store and organize the user's experiences, which we apply for building memories from perceptions and for user support. In the sequel, we describe how the user may apply the artificial memory in order to configure the user support. After a description of related work we conclude with a summary of our results and an outlook of future work.

SCENARIO

Our demo scenario is about a user preparing her shopping trip at home using the World Wide Web, moving to a real world shop, and executing actions in the shop like inspecting and comparing multiple products. Back at home, the user reviews her shopping trip with assistance of the SPECTER system and provides additional information where necessary. This information may be provided by the user on her own free will, or may be requested by the system (e.g., to gather feedback about a service which was suggested by the system, but was rejected by the user).

Multiple sensors are used in this scenario: At home, a special proxy software [13] observes the user browsing the WWW and especially e-commerce sites like Amazon. In the real world shop, the user's actions are recognized by RFID-equipped shelves and products. Additional context information is acquired through web services. Currently we are using weather information and detailed product information (provided by [1]). The only sensor owned by the SPECTER system is the location sensor, which is based on a hybrid system using GPS outside and IR transmitter and active RFID tags inside buildings (cf. [4]).

The system is involved in the described scenario in diverse ways. We address in this article two of them: The recording and analysis of the user's experiences during her shopping trip, and the application of this information for triggering services as part of the user support. Such services may range from management of advertisements over assistance in a product comparison to suggestions for a coffee break. The foundation of this mechanism is a binding between services and situations, which is defined by the user in a collaborative process with the system. This issue leads to another topic addressed in a later section—the question of how a user may specify situations by using the system's memories without being overwhelmed by the sheer amount of recorded information.

INFORMATION ACQUISITION

For the acquisition of relevant sensor information we take a rather practical approach. We expect sensors to publish their information as some kind of location-based service within the area they can "observe". Following this idea, an instrumented shelf would for instance publish its sensor information to devices located in front of that shelf. In principle, the granularity of this approach depends only on the resolution and accuracy of the positioning sensor. As the focus of our project is on the memory structures, we used a slightly simpler approach in our demo implementation based on a hard coded sensor registry published on the local (wireless) subnetwork.

Out of the potentially manifold information sources present in an environment, we consider by default only those providing information with a well defined and machine understandable semantic. This especially includes information represented as instances of concepts defined in some ontology (discussed in more detail in the next section) and in general excludes audio and video data. This is due to the circumstance that we later want to apply automated memory processes on the incoming information, which is hard to do without defined semantics. However, the user could choose at any time to manually add audio or video information to her records.

The sensors in the environment support two modes for information access: Pull and Push. When entering a new context, the SPECTER system first acquires the current status from the newly discovered sensors by use of the pull mechanism. Subsequently, the push mechanism is applied to notify the SPECTER system about observed changes and events in the environment.

MEMORY MODEL

In the following we will describe our application framework with a focus on the memory model responsible for the recording of and the reasoning about the user's experiences.

An overview of our framework and the employed memory model is given in Figure 1. In general, data provided by an instrumented environment is at first collected in a short-term memory to form a snapshot of the user's current context. Support may be delivered in this stage by firing context-aware service triggers previously defined by the user. As the users moves on, outdated information stored in the short-term memory is transferred into the long-term memory. The content of the long-term memory can later be reviewed and evaluated by the user in a process called introspection. The long-term memory provides support based on a user model learned from the evaluated long-term memory content. As such the model is supposed to reflect the user's general attitudes and preferences.

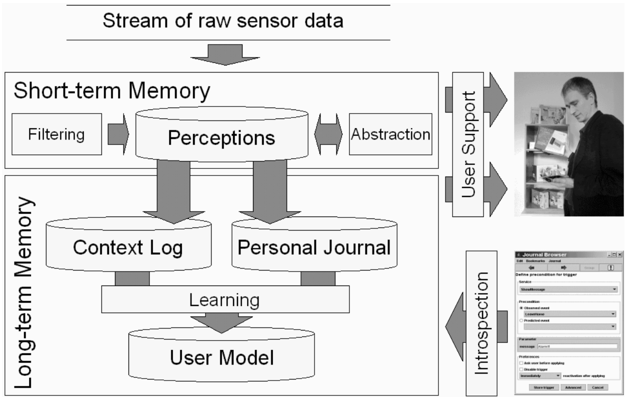

Figure 1: SPECTER's memory model: from low-level perceptions to introspection and user support

Our design was guided by psychologists' research (cf. [10]) on the structure of the human brain and memory. Similar to an artificial system like SPECTER, the human brain has to make sense out of an overwhelming amount of sensor information delivered by the human senses. Obviously, performing sophisticated reasoning based on such low-level information is impractical due to the sheer amount of data to process. Thus, the human brain is organized in different stages, which successively perform information reduction and abstraction.

On the lowest level, the so-called *sensory registry* is responsible for collecting and short-term buffering of basic perceptions. This includes conscious perceptions as well as unconsciously made ones, like for instance the last few words of a conversation we can hear but are not paying attention to. In the human brain, the sensory registry has two main purposes: On the one hand, perceptions arriving at the sensory registry may trigger reflex actions even before we get conscious about them. This may save valuable time and lower the overall cognitive load. On the other hand, if something unexpected is happening, we can reconsider the situation as a whole by incorporating previously recognized but ignored perceptions that would have been otherwise lost.

Both are properties useful for a system like SPECTER: Service triggers occurring in certain situations and contexts can be seen as reflexes of the system, while a short-term but rich history of perceptions is useful to interpret and understand newly arriving information. For these reasons, the first part of our memory model is organized similar to the human brain's sensory registry stage.

In the next stage of the human's brain, perceptions of the sensory registry are transformed into more abstract experiences and perceptions by memory processes, and are stored in the human's short-term memory. This is reflected within SPECTER by an abstraction process we will explain later in this paper. Because there is a close interaction between the sensory registry and the short-term memory, we pooled both in the first stage of our memory model, called SPECTER's artificial *short-term memory*.

After some time in the human's short-term memory, experiences are transferred to the long-term memory stage where they are linked with previous experiences. That way, general attitudes and preferences are established (often we like or dislike something without exactly knowing why), and experiences are related to similar ones which helps to recall them later (the smell of suntan lotion makes us think about our last holiday).

Once again, both are features relevant for a ubiquitous assistant application like SPECTER. Therefore, SPECTER's *long-term memory* was designed in order to allow similar exploitation of recorded experiences. In the first step, experiences from the short-term memory are transferred into a context log storing the plain observations. Additionally, personally coined references between items of the context log are established in the *personal journal.* This in particular includes but is not limited to assignments of favor (user likes or dislikes an experience) and relevance (an experience was considered to be more or less important by the user). These assignments are made either by the user's direct feedback during recording of the observations (for instance via biosensors), or later during an introspection phase. The context log on the one hand allows to link and recall experiences with respect to certain contexts by temporal correlation. On the other hand it serves together with the personal journal as knowledge source for the learning process, which builds and updates a *user model* capturing the user's general attitudes and preferences. Like the personal journal the content of the user model may be reviewed and refined by the user during the introspection process.

Although our design decisions discussed above have been guided by the structure of the human brain, it is important to note that our goal is *not* to build an exact copy of the human brain. As we want to augment and complement the user's memory, there are also fundamental differences to the human brain. The most important one for instance is, that filtering in our short-term memory is by far less restrictive than in the human brain. In our artificial memory, we are trying to gather and store as much information as possible, even if it does not immediately seem to be relevant. That way, we would be able to perform a more in-depth analysis of experiences when required at a later point. For the same reason, at the moment no memory process like the act of forgetting exists in our model. However, older experiences may be assigned a decreasing relevance in reasoning processes in the course of time.

Implementation Details

In this section we want to give details about the current state of our implementation. As we are reporting about ongoing work, the functionality described above has not been fully implemented yet. Therefore, we will focus on the modeling of perceptions, how we store and access them, and what mechanisms we used to implement memory processes.

As a central part of the SPECTER system is the tight cooperation between the user and the system, information needs to be processed in a format meaningful to both. Therefore, we decided to model perceptions and memory entries as instantiations of ontology concepts, based on the IEEE SUMO and MILO ontologies (cf. [12]) with domain-dependant extensions. The main idea is, that each observation made by a sensor forms a self-contained OWL model derived from the underlying ontology classes.

Inside the memory, these perceptions are stored in so called *RDF stores* (with one exception explained later). An RDF store is a persistent collection of arbitrary RDF models with a flexible interface to query and retrieve a collection of models similar to the RDF Net API (cf. [15]). For each model in the store, additional meta information like the source of the model and a timestamp is added. We implemented these RDF stores using Java and the Jena toolkit (cf. [8]). The most important RDF store in the memory model is the context log, which is responsible for the long-term storage of all models of recognized perceptions. The intuition is, that for every type of observation (determined by its ontological class) a virtual "track" exists in the context log. That way, the context at a given point in time can be reconstructed by taking a snapshot of all tracks active at that time. On the other hand, because model content in an RDF store is indexed, all time points with a certain context constellation can be easily identified which is useful for recalling past situations.

The last component we want to describe is SPECTER's memory processes responsible for the transfer of data between different parts of the memory. To implement these processes, we decided to use the JAM planning system (cf. [7]). Doing so, we can define memory processes on a logical level as control strategies working directly on the OWL models of observations and memory items. Thereby, the planning system's fact base is tightly coupled to the respective RDF stores of the preceding memory components. One exception is SPECTER's artificial sensory registry, which is optimized for high throughput instead of long-term storage and is therefore directly implemented by the fact base of the responsible planning process.

Now that the implementation (as described above) has been completed a few weeks ago, we are starting to experiment with different control strategies. Unfortunately, it is to early to present results today. In general our idea is to use a relatively small set of predefined strategies and learn additional rules over time based on user feedback through machine learning.

TRANSPARENT USER SUPPORT

While the short-term memory and the personal journal serve the purpose to store intermediary data and retrievable episodes, respectively, the *user model* (UM) is meant to represent the user's long-term preferences, interests, and goals. This in-depth knowledge about the user is required to enable the system to provide adaptive support, for example by proactively presenting relevant information or triggering (Web) services that meet the user's expectations.

In order to react appropriately, the system must be able to recognize classes of situations and associate these with the activities that are beneficial to the user. Such classification models for situations are derived from lower-level features using a variety of machine-learning techniques. Our current implementation uses decision trees and Naive Bayes.

For a truly ubiquitous system like SPECTER that affects the user's daily life, trust is an important issue. Therefore, the transparency of central processes is an indispensable prerequisite for the acceptance of such a system. Only this way can the system make the user build trust into its mastery of her preferences and, thus, increase the user's acceptance of the overall system behavior (see [2]). This particularly applies to all processes dealing with the acquisition of the UM—such as deriving additional features from sensor data or hypotheses about the user's characteristics—as well as those processes actively using this information to steer the system behavior.

In the attempt to find an acceptable tradeoff between powerful user control and the inherent burden of growing complexity, we designed an intuitive interface that allows the user to interfere even with complex machine-learning processes without the need to deal with technical subtleties of feature selection or data encoding (cf. [3]). The central idea of our approach is to combine the system's capability to deal with *statistical* relevance of a situation's features with the user's ability to name *semantically* meaningful concepts that can and should be used to describe her decision making.

Assume the system tries to create a model that classifies situations according to whether or not a certain service should be executed. For instance, in our shopping scenario the system tries to predict whether or not the user should be presented an advertisement of a nearby store.

In this situation the system will propose a candidate decision tree based on information gathered from the context log and the personal journal. The labeling of training instances stems from user feedback given as a reaction to the system's behavior in previous situations. In order to hide the whole complexity of the classification model, the system only presents what concepts were used to describe episodes from the user's history and discriminate between the two types of situations. Communication between user and system is further facilitated as the user is only shown higher-level semantic features taken from a domain ontology and containing human understandable concepts. The user can then remove (semantically) irrelevant features from the system's list and replace them by other, semantically related concepts taken from the same ontology.

Navigation through the semantic neighborhood of a criticized feature is supported by the system using either a graphical or a list-based interface (see Figure 2). Then the system will re-encode the training data (using a number of heuristics hidden from the user) taking into account the user's specification and iterate until the user is satisfied with the result.[1]

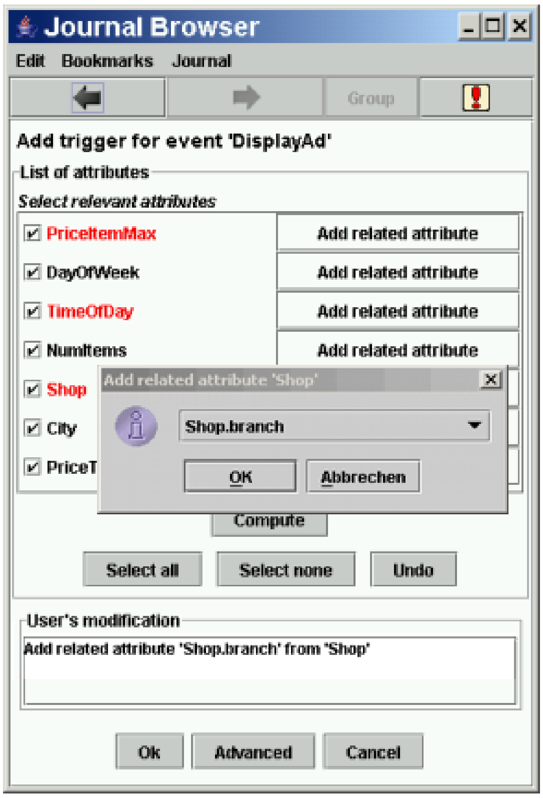

Figure 2: User interface for criticizing and "semantic adjusting" of features selected by the machine learning system due to their statistically relevance.

RELATED WORK

The work described in this contribution is related to several research areas. One central idea is creating an artificial memory, an issue that was the subject of related research, which differed widely in approach and nature of the created memories. In 1994, Lamming and Flynn created a log from sensor input, and pointed out how context information could be applied as a retrieval cue for recalling events in the environment [11]. The permanently growing storage media in mind, Gemmel et al. suggested to digitize the documents created during one's life in order to create a kind of document-centered memory [6]. An example of a product, which has recently appeared in this area is the Nokia LifeBlog software, where data are collected from a camera-equipped cell phone, and are stored in a long-term diary (see *http://www.nokia.com/lifeblog*).

[1] We are currently carrying out a user study to identify the best way to convey the information contained in a decision tree to a naive user.

These projects illustrate the general interest in collecting and filing data, and demonstrate approaches for domains, where rich content is available. However, such input is not necessarily provided by a sensor. Thus, in order to obtain meaningful information from input such as GPS or video, one has to perform an abstraction process in some way. For instance, in [5] clustering of multimedia data is performed to create a diary of situations (e.g., "at the office"). For recognizing basic user states (e.g., "sitting") from acceleration data, in [9] Bayesian classification is employed. An alternative approach is discussed in [14], where objects involved in an activity are mapped to an activity structure mined from the Web.

The memory creation process in SPECTER includes some of the previously mentioned ideas, adapted to the project's specific requirements. These include the collaboration with the user (e.g., to actively collect feedback in response to ambiguous input), and the provision of a mechanism which lets the user add value to the memories beyond archiving.

CONCLUSION

In this paper we sketched the architecture of a system implementing personal, situation-aware, ubiquitous assistance. In order to achieve this goal, the system compiles a kind of memory of observed events comprising aspects of both short-term and long-term memories. The *personal journal* is a kind of episodic memory that can be browsed for interesting events of the past and forms the basis for adaptive user support in a variety of situations.

Machine learning techniques are used to extract relevant patterns from that memory, and thus the user's observed past. These patters allow the system to proactively initiate certain system activities when a particular kind of situation is anticipated or recognized by the system. The collection of these top-level abstractions of the original sensor data forms the core of the *user model* that reflects the user's preferences and expectations in certain classes of situations.

The user largely remains in control over the system behavior and the way it uses her personal data without being forced to engage in lengthy dialogs or deal with complex technical issues. Current work includes the evaluation of certain system aspects w.r.t. usability issues and the integration of various types of sensors providing the low-level input. Future work will be devoted to testing the overall system in complex, mobile scenarios involving a variety of users and services.

ACKNOWLEDGMENTS

Research on SPECTER is being sponsored by the German Federal Ministry for Education and Research (BMB+F under contract 524-40001-01 IW C03.

REFERENCES

1. Amazon Inc. Amazon E-Commerce Service, *http://www.amazon.com/gp/aws/landing.html,* 04/13/05.
2. Bauer, M. Transparent user modeling in SPECTER. *Proceedings of the Seventh International Conference on Work with Computing Systems*, Kuala Lumpur, Malaysia, 2004.
3. Bauer, M., and Baldes, S. An Ontology-Based Interface for Machine Learning. In John Riedl, Anthony Jameson, Daniel Billsus, and Tessa Lau, editors, IUI 2005: International Conference on Intelligent User Interfaces, pages 314–316, New York, 2005. ACM.
4. Brandherm, B., and Schwartz, T. Geo referenced dynamic Bayesian networks for user positioning on mobile systems. *Proceedings of the International Workshop on Location and Context Awareness*, Munich, Germany, 2005.
5. Clarkson, B., Mase, K., and Pentland, A. The Familiar: a living diary and companion. *Proceedings of Computer-Human Interaction*, 2001.
6. Gemmell, J., Bell, G., Lueder, R., Drucker, S., and Wong, C. MyLifeBits: fulfilling the memex vision, 2002.
7. Huber, M. JAM: A BDI-theoretic mobile agent architecture. *Proceedings of the 3rd Conference on Autonomous Agents*, pages 236–243.ACMPress, 1999.
8. JENA – A semantic web framework for Java, *http://jena.sourceforge.net/*, 04/14/05.
9. Kern, N., Schiele, B., and Schmidt, A. Multi-sensor activity context detection for wearable computing. *Proceedings of the European Symposium on Ambient Intelligence*, volume 2875 of LNCS, pages 220–232. LNCS, November 2003.
10. Klatzky, R.L. *Human Memory: Structures and Processes*. W. H. Freeman, San Francisco, 1975.
11. Lamming, M., and Flynn, M. Forget-me-not: intimate computing in support of human memory. *Proceedings of the FRIEND21 Symposium on Next Generation Human Interfaces*, 1994.
12. Niles, I., and Pease, A. Towards a standard upper ontology. *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9. ACM Press, 2001.
13. Obendorf, H., Weinreich, H., and Haß, T. Automatic support for web user studies with SCONE and TEA. *CHI'04: Proceedings of the Conference on Human Factors in Computing Systems*. ACM Press Wien, Austria, 2004.
14. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D., and Haehnel, D. The probabilistic activity toolkit: Towards enabling activity-aware computer interfaces. Technical Report IRS-TR-03-013, Intel Research, Seattle, 2003.
15. World Wide Web Consortium (W3C), RDF Net API, *http://www.w3c.org/Submission/rdf-netapi*, 04/14/05.

Author biographies

Michael Schneider holds a Master of Computer Science from the Saarland University. He works as a researcher at the German Research Center for Artificial Intelligence (DFKI). His interests are ubiquitous computing, mobile systems, and intelligent user interfaces.

Mathias Bauer is a research fellow of the German Research Center for Artificial Intelligence (DFKI). He is interested in user modeling and machine learning/data mining.

Alexander Kröner holds a Ph.D. in Computer Science from the Saarland University. Currently he is working as a senior researcher at the German Research Center for Artificial Intelligence (DFKI). His field of experience comprises the constraint-based creation of multimedia presentations and the application of semantic web technology. In the project SPECTER, his major interest is on the organization and presentation of information stored in the so-called personal journal, a kind of personal user diary.