

Configuration and Introspection of Situated User Support

Stephan Baldes, Alexander Kröner, Mathias Bauer

German Research Center for Artificial Intelligence

66123, Saarbrücken, Germany

{firstname.lastname}@dfki.de

Abstract

User support which takes into account the user's current context is a key issue of ambient intelligence applications. In this contribution we address the question of how a user may configure and keep control of situated user support. Background of our work is a system, which creates an augmented, personal memory from events perceived in an instrumented environment. We show how the user may –assisted by the system– perform a collaborative introspection of this memory in order to learn about and to exploit past events for user support. In particular we discuss methods for evaluating events and for acquiring proactively feedback in order to improve the quality of a binding between situations and system services.

1 Introduction

The notion of ambient intelligence describes a vision of a smart environment, which is aware of users and assists their actions unobtrusively. In order to adapt automatically to the users' needs in a given situation, such an environment has to recognize situations and to decide about means of support suited for the user's context.

Such situated user support requires knowledge about the distinguishing features of a situation. In a general scenario, these may turn out not only to be diverse, but also to depend on the user's personal opinions. Since these may change with time, it is doubtful that there will be "ideal" feature sets in an application with a long-term focus. Consequently one should expect that automated user support will perform undesired actions from time to time, which will affect the user's trust in that support. In order to counter this effect, the recognition of and reaction in wrong decisions become questions of special interest.

We assume that their solution will usually require at least some user feedback about relevant features of a situation on the one side, and about the quality of the system's support on the other side. This leads to the question how such feedback can be acquired unobtrusively and without confronting the user with the full complexity of the problem.

This article contributes to this research area with a collaborative approach for binding user support to general contexts. Our particular focus is on a collaborative *introspection* of augmented memories, which enables the user to review and to adjust the quality of situated user support. Here we think of introspection as "...a process of inward

attention or reflection, so as to examine the contents of the mind..." (cf. [Sharples *et al.*, 1989]).

Our works are conducted within the project SPECTER. Major goals of this project are to extend the user's perception, to learn about her behavior, and to assist her during decision making by means of a personal memory created from user actions observed in an instrumented environment. In this article we will focus only a subset of these works; for an overview see [Kröner *et al.*, 2004].

The remaining article is structured as follows. In Section 2 we go through a scenario in which a user sets up and criticizes situated user support. In the following Section 3 we will discuss the structure of the memories we want to analyze. Then we proceed in Section 4 with a detailed description of our work on the introspection process. After a brief discussion of related work the article concludes with a summary and an outlook of future work.

2 Scenario

In order to clarify our approach to situated user support we continue with an example from the shopping domain.

Our user is interested in buying a new cellphone. She has no precise idea which one to buy, and thus starts browsing an Internet store for cellphones. These actions are logged by SPECTER in her personal memory. At first sight no offer seems appealing to her, and thus she decides to continue her search later. However, in order to facilitate future investigations, she instructs SPECTER to assist her with a comparison between products she is looking at and the ones she has already seen.

Later, equipped with her PDA our user strolls through the city. She planned to buy a specific CD in a consumer electronics store, when she notices that there is also a variety of technical products for sale. She picks a cellphone from the shelf, and thus the criteria that trigger SPECTER's support are fulfilled. On her PDA a product comparison shows up, which compares the features of the phone in the user's hand with the ones of previously inspected phones. This way she browses the shelf, but finally she decides to continue her search later in the Web, and to apply there the information about the seen products. Thus she proceeds to a shelf with CDs and chooses the CD she decided to buy earlier. However, again the comparison's precondition is met, and thus a comparison between this and previously seen CDs shows up. Which doesn't correspond to our user's original intention, and thus she sets a reminder point (a kind of bookmark in the memory) to deal with that problem later.

Home again, our user wants to examine the cellphones seen so far. Thus she starts an introspection of her per-

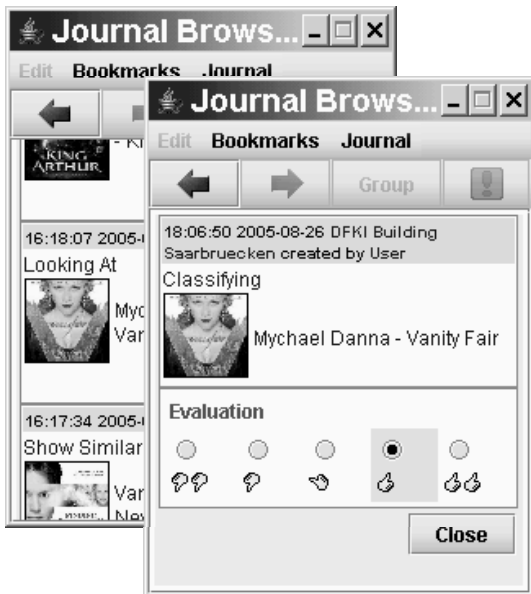


Figure 1: Example views of SPECTER's journal browser: in the background a list of journal entries and in the foreground a rating view.

sonal memory. While she browses through the memories, the system notices the reminder point. When the user has finished her product review, SPECTER asks her if she is now interested in reviewing the reminder point. She agrees, and in response the system offers several actions related to entries around the reminder point. Included is the option to evaluate the CD comparison, which was initiated by the system. She evaluates the service negative, and in response the system asks if she would like to adjust the comparison's application condition. She agrees, and the system offers an automated refinement or a manual adjustment.

3 About Memories in SPECTER

In the scenario described above user support relies on a personal memory created from user actions. Due to their relevance for the situation specification and recognition we will briefly describe in the following this memory's content and structure. For a more detailed discussion we like to make reference to [Schneider *et al.*, 2005].

The personal memory maintained by SPECTER is created from sensor data perceived via a variety of sensors such as RFID, IR, and GPS. In addition, web services as well as a web browser can be connected in order to augment the system's monitoring capabilities. These sensor data are transformed into *perceptions*, which are RDF/OWL-based abstractions from the original data; the ontology applied here is based on the IEEE SUMO and MILO [Niles and Pease, 2001]. In our scenario perceptions include primitive statements such as "user reaching shelf" as well as simple abstractions such as "user holding product". At a given point of time the set of all available sensors' latest perceptions marks the *context* of an event.

Recent perceptions are stored in a short-term memory, which is of special relevance for the recognition of situations. In addition they are archived in a long-term memory. This process is performed in two ways in parallel. First of all, perceptions are stored in a log without further change. This log works as the system's memories. In addi-

tion a *personal journal* is created, which consists of *entries* representing episodes created from one or more perceptions. For instance the perceptions "user holding A" and "user holding B" could be combined to the episode "user comparing A and B". The creation of episodes is an abstraction process, which is performed using pre-authored rules expressing organizational knowledge (which perceptions should be combined) and domain knowledge (which higher-order action is inferred).

The personal journal's main purpose is to provide the user with a higher-order means of accessing the system's memories. In the course of an introspection the user may interact with episodes in varying ways, e.g., by assigning ratings in order to express a personal impression related to a situation, or by applying them in order to customize SPECTER's user support.

4 Introspection of User Support

In our vision SPECTER observes and assists unobtrusively the user during her daily life. User and system communicate by means of a mobile device, which provides access not only to services offered by system and environment, but also to the personal memory. This way the user may use spare time (e.g., during a ride with the train) for an introspection of recent events at any place. The user interface applied in this scenario relies in the first place on the so-called *journal browser*. This component lets the user query, browse and evaluate events and information mentioned there (see Figure 1). SPECTER's support can be customized on a high level (services can be disabled, be configured to inform the user before execution, etc.). Finally, by setting *reminder points* the user may indicate the need for an extensive introspection at an appropriate time (cf. Section 4.2).

Due to resource restrictions in such a mobile setting this kind of introspection is somewhat limited. An extensive introspection can be performed on a desktop PC using a so-called *introspection environment*. This environment provides various ways of accessing and exploiting the personal memory. Its main components are a virtual character, a presentation screen and the journal browser (see Figure 2, from the left to the right). In this environment the user may choose between three modes of performing an introspection.

The *browser mode* focuses the browser-like interface known from the mobile application. While the rich resources of the desktop environment (e.g., the large display) allow the journal browser to provide a more concise interaction than on a PDA, this mode is nevertheless meant for experts. The user is on her own while navigating through the personal journal, evaluating events, exploring the ontology or modifying the user model. A menu bar on the top provides her with direct access to services related to the introspection process. These include general services addressing the personal journal (e.g., querying) and domain-specific services. An example from our shopping scenario is the option to list and evaluate recently made purchases.

In the *mixed mode* the user has again access to the journal browser, but is guided through the introspection by a virtual character. It offers services for the exploitation of recorded memories such as retrieval of additional information about mentioned objects, and serves as the system's "voice". The latter aspect is important if the system has to acquire information proactively from the user, e.g.,

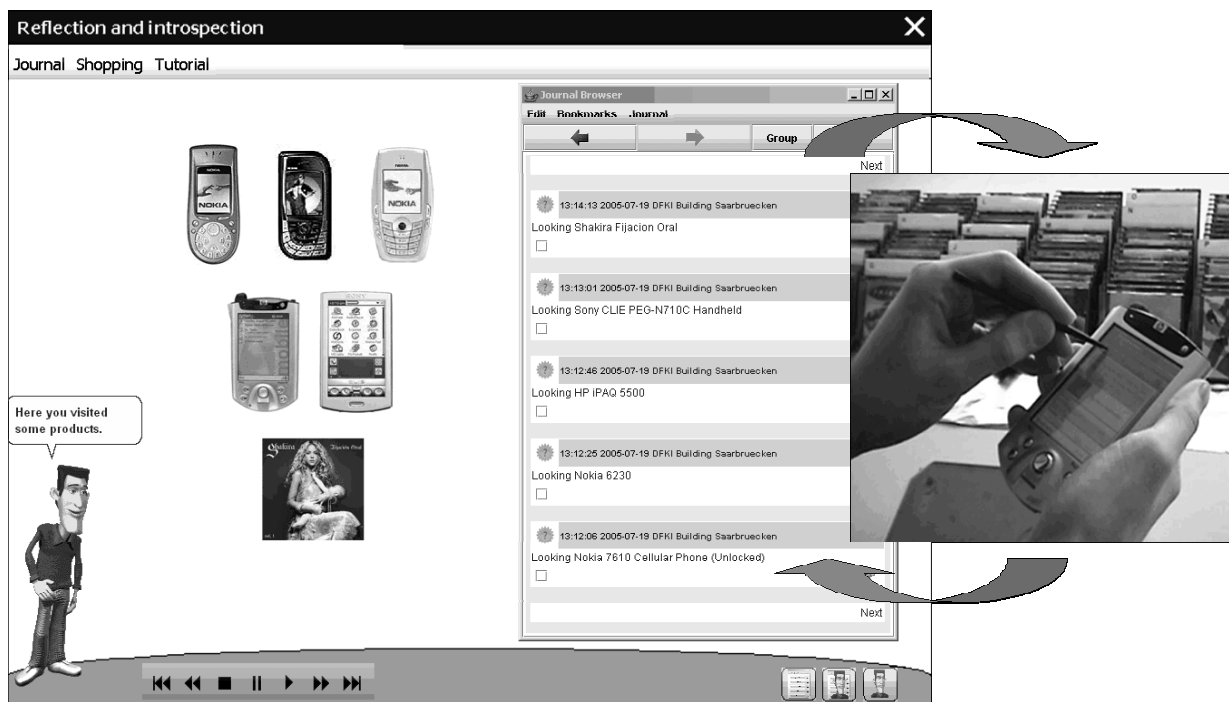


Figure 2: The introspection environment assists the user in reviewing personal memories and in exploiting these for user support in an instrumented environment.

in order to adjust user support after a bad evaluation of a system service. The character communicates by means of audio output, animated gestures and facial expressions. It may display additional information about objects mentioned in the personal memory on the presentation screen, and uses interaction elements such as selection menus and buttons for interacting with the user. A group of control elements allows the user to influence the course of presentation, such as putting in a pause, changing the degree of detail, or omitting parts.

The *character mode* abstracts the most from the underlying technology and aims at an entertaining introspection with little or no interaction from the user's side. The basic idea is to let the character report about recent events. Thus the character appears in the middle of the screen and talks to the user in a conversational tone; components which require explicit user actions are omitted. Additional information appears on the presentation screen if it is crucial to understand the character's remarks.

The introspection process develops according to the chosen mode. Except of the browser mode in which the user interacts autonomously, an interactive presentation is planned at runtime. For realizing the planning and the character's appearance and behavior we integrated the *MIAU platform* [Rist *et al.*, 2003], which makes use of the BDI-Planer *JAM* [Huber, 1999]. Within this platform we modeled the SPECTER system and the character as two autonomous agents with their own planning processes.

The system agent receives events from the environment such as user actions from the browser interface or perceptions from sensors. With this input it plans the top-level behavior of the system such as the automated execution of services or the abstraction of perceptions to personal journal entries. The planer of the system communicates with the character planer and sends instructions, e.g., to start the presentation in a particular mode or to suspend or re-

sume the presentation. In return the character broadcasts its current state to the system.

The character possesses his own strategies to control the introspection process. These consist of general strategies which determine the structure of the presentation, and specific plans which realize tasks required for varying introspection modes. These plans take the current domain into account, e.g. "shopping" in our scenario.

On the basis of the scenario described in Section 2 we will now describe selected elements of a mixed mode introspection.

4.1 Summary of the Day

Let's assume the user has started the introspection environment on the PC. She selects the mixed mode and thus character and journal browser appear. The character greets the user and—because it is already evening—he suggests to review the day. If the user agrees then the character offers a summary of events perceived during that time period. In order to avoid a boring read out of the underlying personal journal entries, the system analyses the personal journal according to domain-dependent plan strategies. It selects events which might be interesting for the user and combines coherent entries to groups which the character verbalizes. In our case he mentions that the user has surfed an Internet store and entered a consumer electronic store. Depending on the selected degree of detail the character also recalls the areas which the user visited in the store, such as the electronics area, music area and cash point area. The fact that the user picked several products from shelves is summarized in one sentence. Because the user is in the mixed interaction mode, events and their temporal context are displayed in the journal browser; in addition images of the visited products are shown on the presentation screen. Figure 2 illustrates such a presentation action.

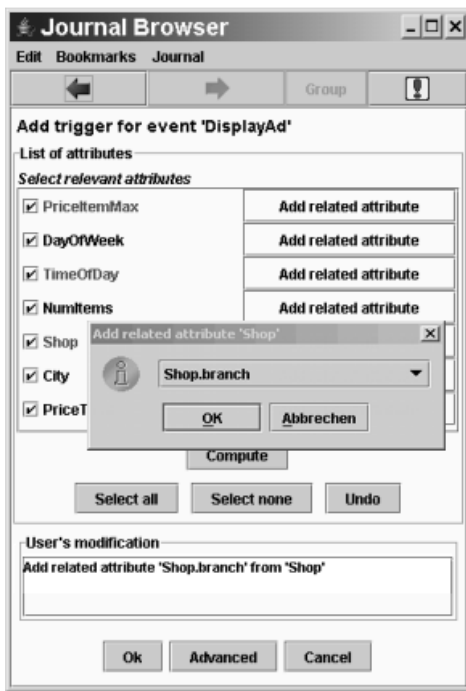


Figure 3: The user interface for critiquing features selected by the machine learning system due to their statistically relevance.

4.2 Reminder Points

Reminder points have a function comparable to bookmarks in time. They mark situations the user wants to review during the introspection. When an introspection is performed on a PC, the system checks if reminder points exist which the user has not reviewed yet. In that case the system asks the user if she wants to inspect these events now. If she agrees then SPECTER steps through one reminder point after the other. Each one is shown in its temporal context. Then the system tries to figure out which of the journal entries created in that context could have attracted the user's interest, and offers services selected according to this entry's content. In our example the user set a reminder point after a product comparison appeared. SPECTER guesses that the user wants to review this action and offers special services linked to the comparison such as showing it again on the larger screen, or acquiring product descriptions and customer reviews of the compared products from the Internet (on the implementation side, these services rely on web services provided by Amazon.com). In addition to these product-centered services the user can also select general services such as reviewing or evaluating the events discovered in the reminder point's neighborhood.

4.3 Proactively Improving the User Model

While SPECTER supports the user even without explicit feedback, the quality of this support will usually increase if the system knows about the user's personal opinion regarding the system's behavior. Therefore SPECTER proactively asks for feedback in the form of negative or positive ratings of events created in response to system actions. Like most introspection actions such queries have an optional character. Thus the user is not obliged to pro-

vide feedback; she can postpone it to a more convenient time or dismiss the query at all.

In our scenario the user is willing to invest some time. Hereon she is asked to evaluate the appearance of the product comparison, which was automatically invoked by user-defined service binding rules. The user is provided with a list of journal entries making reference to the product comparison; each of these entries has attached a link to a rating form. In our example she rates the CD comparison negative and the cellphone and PDA comparisons positive. Thereupon the system suggests to change the parameter setting for this service binding. It asks if the user wants to do this manually or if she wishes to hear an recommendation from the system. The recommendation method tries to eliminate the cause of negative rated system actions.

This method is realized as a system plan and depends on the type of the negative rated action. In the case of an automatic service execution the system tries to identify which parameter settings of the user-defined service binding are rated as negative. With aid of the underlying ontology it proposes alternative settings which reduce applications of services in negative rated situations. Following our example, the user asks for the system's advice. It suggests to specialize the parameter "products" from "general products" to "technical products", because cellphones and PDAs are technical products and are rated positive whereas non-technical products as CDs are rated negative. The user accepts this recommendation and the system performs this modification of the user-defined service binding by itself.

4.4 Collaborative Critiquing the Service Binding

If the user is not confident with the system's suggestions—or if she wants to set up a new service binding—she can proceed to a component explicitly designed for mapping situations to services. This component is described in detail in [Bauer and Baldes, 2005]; at this point we will only summarize its features and focus on its role within the introspection process.

The goal of this component is to provide a scrutable and easy-to-use interface that allows the user to interfere with complex machine-learning processes without the need to deal with the technical subtleties of feature selection or data encoding. 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.

The outcome of our ongoing efforts is a user interface, which provides several layers with varying complexity for combining services and situations. Especially relevant for the critiquing a situation's features (and thus for configuring the execution of bound services) is the screen shown in Figure 3. There a list of features is presented to the user, which have been extracted using statistical methods from journal entries. The user may now criticize this set by deselecting features, or by navigating in the semantic neighborhood using a graphical interface to the underlying ontology. Thus in our example scenario the user might want to inform the system that the product comparison should only be applied in certain kinds of shops. A way to achieve this goal is to inspect the feature list and then to refine the shop's features, e.g., by replacing the general shop by a more specific class.

5 Related Work

The work described in this contribution is related to several research areas. Its central idea of creating and exploiting augmented personal memories is the subject of various research efforts. For instance, an augmented memory created from automatically captured photos and GPS data is described in [Gemmell *et al.*, 2005]. An introspection-like process lets the user author presentations from these memories. The introspection of event histories is discussed in [Horvitz *et al.*, 2004]. There the user's desktop communication is organized on a timeline according to learned memory landmarks. A kind of manual introspection is described in [Ito *et al.*, 2004], where visual programming allows the user to define personal memory viewers for analyzing action histories.

Situated user support as described in this article requires an analysis of the user's context. How such an analysis can be performed by means of ontologies and inference is elaborated in [Chen *et al.*, 2004]. A mechanism for deriving high-level contexts like in our approach from environmental events is discussed in [Tan *et al.*, 2005]. Interesting for the fully automated optimization of user support is the approach described in [Castro and Muntz, 2000]. Here evidential reasoning techniques are applied in order to extract and infer useful context information from sensor data. Our contribution to this area is a collaborative process, in which aspects of such systems (such as ontologies, inference and machine learning) are combined in order to enable the user to inspect and adjust situated user support.

6 Conclusion and Outlook

We reported in this article about an approach to the configuration of situated user support in pervasive computing environments. Based on an episodic memory created from automatically perceived user actions, we showed how in a collaborative introspection process system and user together adjust bindings between situations and system services. This process involves methods such as reminder points for making notes of situations, an ontology-based analysis of elements referenced by situations, a guided tour through events perceived by the system, and finally a user interface for adjusting and critiquing a binding between situations and system services.

In order to deal with context changes and with new application scenarios this kind of introspection requires further research. Its evaluation is part of an ongoing study, in which various aspects of SPECTER are analyzed.

Acknowledgements

This research was supported by the German Ministry of Education and Research (BMBF) under grant 524-40001-01 IW C03 (project SPECTER). We would like to thank the SPECTER project members for their great support.

References

[Bauer and Baldes, 2005] Mathias Bauer and Stephan Baldes. *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. ACM Press, New York, 2005.

[Castro and Muntz, 2000] Paul Castro and Richard Muntz. *Managing Context for Smart Spaces*. IEEE Personal Communications, Vol. 7, no. 5, 2000.

[Chen *et al.*, 2004] Harry Chen, Tim Finin, and Anupam Joshi. *Semantic Web in the Context Broker Architecture*. IEEE Conference on Pervasive Computing and Communications (PerCom), 2004.

[Gemmell *et al.*, 2005] Jim Gemmell, Aleks Aris, and Roger Lueder. *Telling stories with MyLifeBits*. In Proceedings of the IEEE International Conference on Multimedia & Expo (ICME) 2005, Amsterdam, The Netherlands, 2005.

[Horvitz *et al.*, 2004] Eric Horvitz, Susan Dumais, and Paul Koch. *Learning predictive models of memory landmarks*. In Proceedings of the CogSci 2004: 26th Annual Meeting of the Cognitive Science Society, Chicago, USA, August 2004.

[Huber, 1999] Marcus J. Huber. *JAM: A BDI-theoretic mobile agent architecture*. Proceedings of the 3rd Conference on Autonomous Agents, pages 236–243. ACM Press, 1999.

[Sharples *et al.*, 1989] Mike Sharples, David Hogg, Chris Hutchison, Steve Torrance, Chris Hutchinson. *Computers and Thought: A Practical Introduction to Artificial Intelligence*. Bradford Book, 1989.

[Ito *et al.*, 2004] Masaki Ito, Jin Nakazawa, and Hideyuki Tokuda. *A framework for personalizing action history viewer*. In Proceedings of PERVASIVE 2004 Workshop on “Memory and Sharing of Experiences”, pages 87–94. Vienna, Austria, 2004.

[Kröner *et al.*, 2004] Alexander Kröner, Stephan Baldes, Anthony Jameson, and Mathias Bauer. *Using an Extended Episodic Memory Within a Mobile Companion*. In Proceedings of PERVASIVE 2004 Workshop on “Memory and Sharing of Experiences”, pages 59–66. Vienna, Austria, 2004.

[Niles and Pease, 2001] Ian Niles and Adam Pease. *Towards a standard upper ontology*. FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems, pages 2–9. ACM Press, 2001.

[Rist *et al.*, 2003] Thomas Rist, Elisabeth André, Stephan Baldes. *A Flexible Platform for Building Applications with Life-Like Characters*. Proceedings of Intelligent User Interfaces (IUI) 03. ACM Press, 2003.

[Schneider *et al.*, 2005] Michael Schneider, Mathias Bauer, and Alexander Kröner. *Building a Personal Memory for Situated User Support*. 1st International Workshop on Exploiting Context Histories in Smart Environments, Munich, Germany, 2005.

[Tan *et al.*, 2005] Joo Geok Tan, Daqing Zhang, Xiaohang Wang, and Heng Seng Cheng. *Enhancing Semantic Spaces with Event-Driven Context Interpretation*. In Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005). Springer, 2005.