Customizing the Building and Exploitation of an Event-Based Digital Memory

Catalin Barbu and Alexander Kröner

German Research Center for Artificial Intelligence,
Stuhlsatzenhausweg 3, 66123 Saarbrücken
{Catalin.Barbu, Alexander.Kroener}@dfki.de

Abstract. Enhancing user perception in smart environments is an active area of research. Its foundation is the use of sensor technology in order to capture automatically events that might otherwise escape the user's knowledge. Organized in a personal diary, such events may be exploited for various kinds of user support. However, while the approach requires that the user is not forced to interact with incoming information, the information captured should nevertheless match her expectations. In this article, we describe various strategies that let the user control the creation and structuring of events generated in a virtual shopping environment. Our report comprises a summary of information sources offered to the user in this setting as well as methods employed to construct a diary of shopping actions in a semi-automated way.

Keywords: digital assistant, augmented memory, privacy, customization

1 Introduction

Large-scale advances in sensor technology, particularly RFID, have made it possible to create instrumented locations, in which the interactions between users and the environment can be captured, stored and analyzed as discrete events. By collecting such events, a diary-like structure like the one shown in Figure 1 can be created, which can be exploited by digital assistants in various ways, e.g., to provide support based on the user's habits, or to let her reproduce past actions.

Automated approaches are especially qualified for building such a diary, since they free the user from the burden of taking notes and allow for capturing information which she otherwise would not have noticed. An example of such a system is described in [1], which allows trusted external applications to communicate observations to the user's diary. However, the user has little control over *which* entries a trusted application adds to the diary. This is not necessarily an issue if automated processes make use of the diary, but might be in contrast with the user's expectations of a personal diary. This issue can be addressed via the organization of events. For instance, the authors of [2] employ machine learning in order to generate episodes from large sets of events in a desktop setting. This way, the user obtains an abstract view on the collected mass of information. This process is guided in the very first place by domain knowledge (e.g., about the operating system) and user feedback.

Figure 1: A diary created with SharedLife (screenshot). User actions observed by a store (e.g., "Looking") are combined with actions performed by the store (e.g., "Recommending"). Actions are organized in a hierarchical manner.

In the following, we report on an approach to controlling the creation and the exploitation of diary entries which is used in the SharedLife system. Our contribution is twofold. We show how the diary itself can be used for negotiating the creation and exploitation of diary entries of previously unknown locations with the user. On top of that, we propose a mechanism which allows external applications to make suggestions concerning the organization of entries based on their respective domain knowledge. This work was partially funded by the German Ministry of Education and Research under grants 01IWF03 (project SharedLife) and 01IA08002A (project SemProM).

2 Controlling the Creation and Exploitation of Diary Entries

The virtual shopping environment described in [3] exploits the user's diary in order to provide in-store navigation support and product recommendations. These services rely on information SharedLife extracts from the user's diary. The environment supports this process indirectly via a stream of events, which describe user actions such as "Arriving", "Looking (at product)", or "Leaving". In a similar way, services executed by the store are reported to SharedLife, such as "(Store) Recommending". SharedLife in turn may ignore all of these requests.

These automated processes are controlled by privacy profiles. Once the user has entered a store, it provides SharedLife with a list of diary interactions needed to let the user benefit from all of the store's services. SharedLife then checks if there is a privacy profile defined for this store. If there is none, SharedLife uses its own (mobile) interface in order to mediate the access with the user.

This request is logged in the diary and can be answered by the user at any time. Similarly, the user's confirmation is recorded together with the profile. If a profile allows an external source to create entries in the diary, then these are linked with the respective profile. This tight interconnection of a history of profiles and diary entries provides the user with a high level of control over when such profiles are created, confirmed and deleted – and with an opportunity to learn about their impact: it

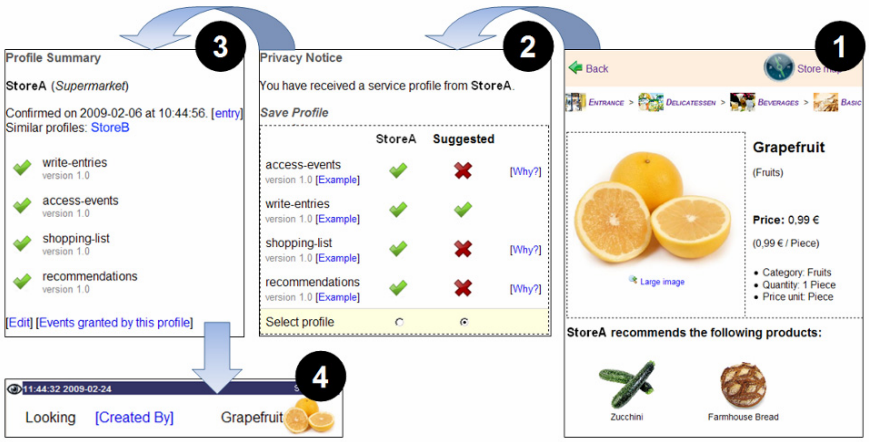

Figure 2: (1) The virtual shopping environment; (2) profile negotiation in SharedLife; (3) confirmed profile; (4) event submitted by the store.

enables the system to answer queries such as "Which profile allowed the generation of this entry?" or "Show me all events granted by this profile".

Consequently, SharedLife exploits this information during the profile negotiation with a user (cf. Figure 2). First, SharedLife retrieves privacy profiles for the same class of location (e.g., supermarkets) from the diary. Next, the system computes a statistics from the retrieved data in order to propose a profile based on the frequency of the user's decisions for that class. If no profiles are found, then a default profile is taken from the memory. This default profile is initially very restrictive, but can be edited by the user in order to suit her needs. The user can accept the profile submitted by the store or the one recommended by SharedLife, or she can define her own privacy settings. Once a privacy profile has been confirmed, the user is still able to edit the privacy settings at any time – even while shopping, or she can block the profile from being used, thereby rendering SharedLife "invisible" to the external location.

3 Structuring Events

Both the user's and the store's actions will quickly fill the user's diary with numerous events. During shopping, the user will not take notice of this automated process, since most user support features select and compile data from the memory into task-specific views such as product and function lists (cf. [4]). However, a digital memory can also be exploited for reviewing past actions – a task which is especially crucial for discovering events not observed by the user. In order to support discovery and exploration of events, we enabled the diary to support the user and external applications with a simple form of abstraction resembled by event hierarchies. In contrast to the mining of episodes in [2], the generation of these hierarchies is not a

feature of the diary itself. This is due to the fact that the diary may be used across various scenarios. For instance, SharedLife was also tested in a cooking scenario with quite different event creation behaviour in terms of number and granularity (see, e.g., [5] for an action model). These differences result not only from the domain, but also from the respective applications: these employ knowledge which is not fully accessible to the domain-independent diary (e.g., algorithms reflecting the respective sensing equipment).

Therefore, we decided to compute hierarchies where the knowledge actually is located: within the applications. These communicate the result of their computation – a set of relationships among previously generated events – as an initial proposal for a hierarchy to the diary. That proposal is treated like a regular event and is therefore subject to the previously described privacy profiles. If granted by the profile, the system combines this information with the diary. For the user, the result is displayed as the tree-like view depicted in Figure 1. The user may customize the result at any time – here manually via drag and drop – if she does not agree with the proposal.

4 Conclusion

For the purpose of user modelling, input from intelligent environments in general needs to be filtered and structured. In this article, we discussed an approach where an event-based digital memory is applied in order to support the user in the definition of profiles, which regulate the handling of events communicated by the environment. In order to organize incoming events, we proposed to allow the environment to exploit its domain knowledge to generate a structure proposal, which – if granted by the respective profile – can be combined with the memory.

References

1. Boehm, S., Koolwaaij, J., Luther, M., Souville, B., Wagner, M., and Wibbels, M. (2008). Introducing IYOUIT. In Proceedings of ISWC 2008. LNCS, vol. 5318, pp. 804–817. Springer: Heidelberg.
2. Horvitz, E., Dumais, S., and Koch, P. (2004). Learning Predictive Models of Memory Landmarks. In Proceedings of the CogSci 2004: 26th Annual Meeting of the Cognitive Science Society, Chicago, USA.
3. Barbu, C., and Kröner, A. (2008). Designing a Study Concerning the Functions of Sharable Personal Memories. In Proceedings of the IADIS International Conference WWW/Internet 2008, October 13–15, Freiburg, Germany.
4. Kröner, A., and Basselin, N. (2006). Situated Views on Augmented Memories. In Proceedings of the IADIS International Conference WWW/Internet 2006, pp. 272–277, Murcia, Spain.
5. Stahl, C., Heckmann, D., Schneider, M., and Kröner, A. (2008). An Activity-Based Approach to the Design of User Assistance in Intelligent Environments. In Proceedings of the AmI 2008 workshop on Capturing Ambient Assisted Living Needs, Nürnberg, Germany.