SPECTER: a User-Centered View on Ubiquitous Computing

Thomas Kleinbauer and **Mathias Bauer** and **Anthony Jameson**
DFKI, German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, 66123 Saarbrücken
firstname.lastname@dfki.de

Abstract

Although there still remain a lot of open questions on technical issues in ubiquitous computing, mobile system must not forget to put the user in the center of their interest. Context data must be presented in an intelligible way, all system behavior must support the user according to a model of her preferences and interests and privacy concerns should always be seen as a key issue. In this paper, we discuss various aspects of these topics on the basis of the SPECTER system, without raising a claim to completeness.

1 Introduction

In recent years, many prototype systems have been developed that include some of the following interrelated properties: awareness of the user's context, in particular during mobile use; communication with smart objects in networked environments; sensing of and addressing the user's affective states; and proactive, context-sensitive information retrieval and decision support. Although considerable progress has been made and some prototypes exhibit impressive performance, there remain several broad challenges that need to be addressed.

Among those, one key problem that arises in just about any context-aware system is to find means of capturing information from the environment. This is not only a matter of signal processing, but also requires effective mechanisms to (a) combine and classify various heterogenous data sources, (b) abstract low-level sensor data to human-readable representations, (c) learn and draw inferences from previous events in accordance to an underlying user model, (d) satisfy the user's sound claim for privacy.

The SPECTER project at DFKI aims to contribute to the challenges of more explicitly modeling and analyzing users' behavior and responses, both during the design of the system and during its use. Thereby, we try to take a user-centered point of view. In this paper, we give a first overview over the topics that our project addresses and also provide some ideas on important issues like context exploitation and privacy.

2 Related Work

The primary goal of the early system FORGET-ME-NOT ([Lamming and Flynn, 1994]) was to support human episodic memory by capturing and storing data about everyday activities and providing mechanisms for the retrieval of such information. One insight underlying this early prototype was that part of the stored information, though unimportant in itself (e.g., the fact that the user received an irrelevant phone call from a sales representative on a given morning) can still serve as a useful retrieval cue for more important information (e.g., "Who was the woman who called me just after that irrelevant phone call?") [Lamming and Flynn, 1994] argued persuasively for the utility of augmenting episodic memory, and their prototype illustrated how several types of context information could be captured automatically: *U*'s [1] location, *U*'s encounters with other persons (identified via their smart badges), their activities at their stationary work station, their use of printers, and their telephone calls. As the authors pointed out, the set of types of data that can be captured is open-ended, and it can vary even from one concrete situation to the next. The authors also supplied a means of representing the database of events with easy-to-scan icons; these icons were also employed by users in the formulation of queries.

A more recent system that captures and graphically represents *U*'s actions and their context is COMICDIARY ([Sakamoto *et al.*, 2001]). This system is part of the C-MAP project, which creates guidance systems for events such as visits to conferences, trade fairs, and museums. COMICDIARY, tested in a conference setting, leverages data that has been captured and stored in a central database on the basis of *U*'s use of the C-MAP guidance systems PALMGUIDE and AGENTSALON (see, e.g., [Sumi and Mase, 2001]). The data include records of events that *U* has visited, *U*'s subjective evaluations of these events (as expressed by explicit ratings), and *U*'s use of the guidance systems. Since COMICDIARY makes no use of this information besides visualizing it, its interest lies in the way it augments the user's episodic memory. The captured data is, however, used in other ways in PALMGUIDE and AGENTSALON.

Some systems have supported efficient mobile note-taking in one form or another and/or capture of raw data such as audio or video streams. One example of this type of data capture is provided by a family of context-aware systems developed at Microsoft Research (see, e.g., [Horvitz *et al.*, 1999]; [Horvitz *et al.*, 2002]). The systems PRIORITIES and COORDINATE capture data about a user's activities (e.g., absence from her work station, or more specific activities at the work station) so as to be able to determine the timing of notifications (e.g., whether to relay an email message to *U*'s mobile device or to save it until *U* has returned to the work station). The sources of information include sensors such as microphones, databases such as *U*'s electronic calendar, and logs of low-level events in

[1] The user of SPECTER will sometimes be referred to with the symbol *U*, while the system will sometimes be referred to as *S*.

U's computer use (e.g., mouse clicks). With regard to the last type of data, this research yields one example of how to approach the problem of recoding low-level events in terms of more interpretable actions and patterns of activity: The same event interpretation methods are applied that were developed for the LUMIÈRE prototype ([Horvitz *et al.*, 1998]), the forerunner of the commercially deployed OFFICE ASSISTANT. The system was able to learn to predict quite well the subjective *interruptibility* of a given meeting for a given *U* (i.e., the extent to which *U* would be willing to be interrupted during the meeting). But since this important property can never be observed directly, the user in their study had to make explicit assessments for 559 meetings in order for learning to be possible. This example illustrates the potential utility of physiological sensors in reducing the amount of explicit input required: Instead of learning to predict "interruptibility", a system might learn to predict affective responses to actual interruptions in meetings.

In previous research on context-aware systems that capture data about users' actions and their contexts, only a minority of the systems have performed serious learning on the acquired data. One recent positive example is the family of Microsoft Research systems: On the basis of extensive data about a given user's activities and locations, these systems learn Bayesian network models for predicting such activities and locations. This research has several salient features: Each model is learned on-line and on the fly, when the system requires a prediction, on the basis of the subset of the available data that the system deems relevant to the prediction. While this approach has the advantage of permitting the learning process to be optimized for each individual prediction, it appears hard to reconcile with the goal of creating models that the user can inspect and modify and whose predictions the user can understand. The designs of the Microsoft systems appear to be based on the premise that ultimately the systems' performance will be so good that users will be willing to rely on the system's predictions and recommendations implicitly. By contrast, the design of SPECTER will be based on the assumption that at least some users will demand some degree of understanding and control—an assumption that has far-reaching consequences for the type of model that the system should learn.

There exist a number of approaches to the problem of inferring behavior patterns from observed user actions. This information is crucial for systems like SPECTER as it constitutes a prerequisite for supporting *U* by pointing out opportunities, identifying relevant resources, or activating external services. [Yoshida *et al.*, 1994] and [Yoshida and Motoda, 1995] describe how a graph representing a UNIX user's typical behavior can be learned from sample action sequences. *Graph-based induction* identifies common patterns in these traces. On the basis of transition probabilities between the various nodes (actions), the user's next command can be predicted. However, only the use of deep domain knowledge can ensure high prediction accuracy. While the notion of *plans* as abstract, goal-related structures plays no role, a system might extract these by following the most probable paths through the graph.

The IPAM algorithm presented by [Davison and Hirsh, 1998] is a *knowledge-free* method for constructing a probabilistic model of action patterns contained in action sequences that is used to predict future actions. This is in contrast to the work of [Yoshida and Motoda, 1995], where information about I-O relationships among UNIX commands is exploited to produce a probabilistic predictive model. Unlike IPAM, this *graph induction* method needs relatively few training data to successfully produce predictions. Both approaches do not explicitly mention plans but rather try to detect regularities in user behaviors in order to predict the user's next actions.

[Albrecht *et al.*, 1997] use dynamic Bayesian networks to map the behaviors of players in a multiuser dungeon to the goals ("/quests"/) currently being pursued. While the set of possible goals can be completely enumerated, the enormous number of actions and places renders any attempt to create a complete domain model futile—a characteristic that can also be found in the dynamic instrumented environments of SPECTER. As a consequence, there is no such thing as an operational description of the various possible ways to achieve a particular goal. Instead, given a set of sample action sequences, the current quest is predicted on the basis of statistical correlations between the previous quest and the current location and action. This method demonstrates its feasibility in a scenario where there is no standard way to reach a particular goal state.

[Bauer, 1998a; 1998b; 1999] introduces an approach based on machine learning techniques that allows abstract plan libraries to be derived from observed low-level action sequences. Using clustering techniques, it is possible to generate characterizations of user behavior on various levels of abstraction. Additionally, it is possible to create alternative plan decompositions in cases where there is more than one way to achieve a particular goal. Depending on the intended use of the plan libraries so created, it might be necessary to include a human expert in the process who can attach semantic labels to the resulting knowledge structures that are otherwise hard to convey to a user.

[Garland *et al.*, 2000] discuss ways to overcome the limitations of these approaches to sequential plan structures. They discuss purely interactive approaches, the use of machine learning methods, and mixed initiative techniques. This work has been followed up on by [Garland *et al.*, 2001]. Here, a supervised learning approach is taken to produce hierarchical task models for the COLLAGEN system. A domain expert is expected to interact with the system, annotating examples and refining the system's hypotheses until a satisfactory description of a task is achieved.

A promising approach consists in applying metalearning techniques that try to classify the past into a series of episodes. Given a current learning task, only considering the most recent data could be insufficient. So the idea is to find situations in the past that are similar to the current one and take into account the data and results associated with them. Doing so typically results in improved predictive accuracy. This approach, which is advocated by [Widmer, 1997] and [Koychev, 2002], is similar to that of the two Microsoft Research systems discussed at the beginning of this subsection.

Context-awareness has often been claimed to be a central prerequisite for intelligent systems that provide assistance to their users in a ubiquitous computing environment. For example, this claim is fundamental to the Intelligent Room project (see, e.g., [Hanssens *et al.*, 2002]). The research is focused on specifying an appropriate communication infrastructure, on aspects of automatic resource allocation, on allowing users and services to share resources, and especially on a reactive behavioral system that not only facilitates building a context representation, but also uses that knowledge to provide relevant services to the user. Since complex behavior models can be built, conflict resolution becomes a central problem (see, e.g., [Shafer *et al.*, 2001]).

An example of a purely individual context-aware system is CYBREMINDER ([Dey and Abowd, 2000]). It is a prototype tool that supports users in sending and receiving reminders that can be associated with richly described situations involving time, place, and more sophisticated specifications of context that may be supported by the infrastructure used (see, e.g., the CONTEXT TOOLKIT of [Salber *et al.*, 1999]). It is intended to augment traditional calendar-like tools.

A more extended use of sensor data is realized in MEMORY GLASSES ([DeVaul *et al.*, 2000]), a wearable context-aware reminder system. It uses time, location, and activity to deliver reminders. It focuses on personal context and uses body-worn sensors (a camera and a microphone) to determine what activity the wearer is engaged in, including walking down stairs or taking part in a conversation.

ery few projects have attempted the more sophisticated user modeling, planning, and decision making that is planned for the SPECTER project. One exception is the recently initiated work of [Byun and Cheverst, 2002], who aim to achieve higher-than-usual levels of context interpretation as a basis for the proactive behavior of their system. They explore the potential of utilising machine learning techniques based on the processing of the current context with respect to the context history and the users' feedback concerning new situations and the explanations that the system gave about its behavior. This work illustrates some of the constraints that govern the selection of learning and decision making methods: Many methods are found to lack the necessary transparency for users.

Closely related to the interpretation of context, especially in an instrumented environment containing smart objects, is research in the field of functional reasoning (see, e.g., [Chandrasekaran, 1994]; [Chandrasekaran and Josephson, 2000]). Whenever a SPECTER system detects a mismatch between the current needs to be satisfied and the functions that the environment provides, *S* should be able to reason about this mismatch on the right level of abstraction.

3 Example Scenario

In order to get a more concrete idea what kind of system we have in mind, imagine the following scene: The user is on a shopping tour through a city. During the last shopping trip, her SPECTER system recorded both her navigation (e.g., using GPS logs) and her shopping behavior (e.g., using online credit card logs). It kept track of the places and shops visited, the (emotional) quality of the shopping experience (e.g., whether or not the sales representatives were friendly and knowledgeable), and also what types of goods *U* actually purchased. Knowing about *U*'s current shopping plans and her previous experience in that city, *S* can recommend shops and guide her to the right places while avoiding possibly unpleasant or dangerous locations. Additionally, *S* actively filters the large amount of advertisements and discount offers broadcast by the various shops using location-sensitive information services. In this way, it protects its user from unwanted and irrelevant messages while detecting interesting offers even from previously unknown shops and providers.

4 SPECTER's System Architecture

Since we're still at the very beginning of our research in the SPECTER project, nearly all of the components are yet to be implemented. Still, we hope that by illustrating our approach and the ideas behind each planned module we are able to contribute to the general field of ubiquitous computing and provide a basis for discussion.

Figure 1 shows an abstract overview of SPECTER's system architecture. SPECTER relies mainly on two knowledge sources, an abstraction of the user's events and actions called *personal journal* and a *user model*. Both components are described in more detail in later sections of this paper.

The square boxes represent those parts of the system that perform some kind of computation. All of them interact more or less directly with the knowledge sources by querying particular information, adding new knowledge, or manipulating existing entries. To do so, the system utilizes different kind of input sources, for instance it communicates with the environment through smart sensors and with *U* either through direct interaction e.g. using the handheld client or through body sensors to infer the emotional and affective state of the user.

In order to fulfill its main goals —support the user by taking over some of her tasks and serve as a episodic memory aid— SPECTER's architecture provides the abilities to call relevant services in the environment (e.g. in the shopping scenario, SPECTER might query the department store for special offers on a specific type of product) and to directly address the user (e.g. that it has discovered a special offer on detergent in the current store).

4.1 Extension of Perception

Since virtually all context-aware systems capture some sort of information from pervasive date sources, many mechanisms have already been successfully implemented and tested in previous research. Consequently, some of the work in SPECTER in this area will largely involve the evaluation, selection, and adaptation of existing techniques. Still, the innovative aspects of SPECTER begin where the captured data about the user and her context are used as a basis for processes of information retrieval and learning (as opposed to the more common triggering of straightforward system adaptations). One research challenge will therefore be to find ways of preprocessing, representing, and storing the captured information that will support such processes.

Perhaps the most complex new research issue in this area stems from the recognition that users must accept some responsibility for the protection of their own privacy and security, in particular by exercising control over the collection and storage of data (cf. section 6). The design of adequately flexible and unobtrusive control mechanisms involves considerations ranging from details of user interface design to assessments of the role and value of particular types of data for particular purposes.

4.2 Learning About Behavior and Affect

The data about the user and the environment that is collected by SPECTER will be used in two general ways:

1. Stored in the personal journal, it will serve as an extension of *U*'s episodic memory. *S* will answer explicit queries by *U* about past events. Moreover, the active decision making component of SPECTER will sometimes spontaneously offer information about past

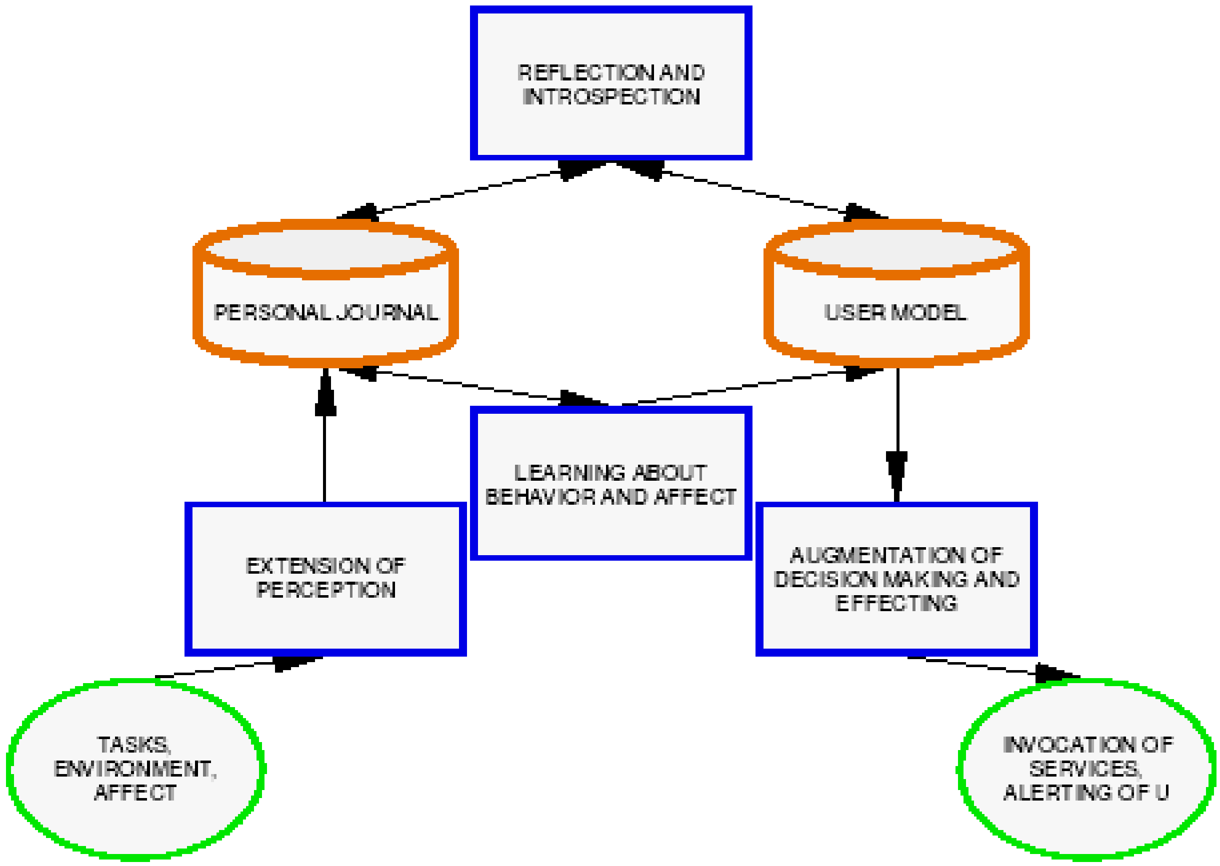

Figure 1: The main components of the SPECTER system

events to *U* if such information appears relevant to *U*'s current or planned activities.

2. On the basis of the personal journal, *S* will learn a more general *user model*. Instead of referring to particular events, the user model represents in a more general form what *S* has learned about *U*'s preferences, habits, capabilities, and other properties that are relevant to prediction and decision making on *U*'s behalf (s. 6).

In order to provide these functionalities, various types of information have to be extracted from the data available. First, it will be necessary to identify *classes of situations* along various dimensions. These will include situations in which *U* was happy or angry, in which she performed particular types of actions, or in which she used certain resources. Such classes can be characterized in terms of features that are perceivable by SPECTER. They will enable SPECTER to provide advice to *U* when a certain situation arises that can be identified as belonging to one of these classes.

Similarly, detecting *patterns of behavior*—characterized, for example, by typical action sequences and resource usage—could serve the purpose of supporting *U* in her attempts to achieve certain goals. In particular, they could be used by SPECTER to recognize favorable opportunities to help *U* to continue working toward a previously suspended goal, for example, when *S* has found a missing resource (see section 4.3).

When trying to extract the required information at an appropriate level of abstraction and with sufficient precision, the problem arises of how to combine manual editing and annotation by *U* and (semi-) autonomous data mining mechanisms. The overall goal must be to put the human into control without requiring too much manual input, while simultaneously guaranteeing a sufficiently high level of correctness of the learning results.

The *scrutability* (i.e., inspectability and transparency) of the resulting knowledge structures will be essential to the acceptance of *S*'s behavior and *U*'s trust in its reliability. In particular, *U*'s modifications to *S*'s hypotheses about her behavior and affect during the *reflection* phase must be mirrored in *S*'s subsequent learning performance. The necessity of providing an appropriate, intuitive interface to *U*'s personal journal—including *S*'s inferred assumptions about *U*'s goals, plans, and affective states—will be discussed in section 4.4.

Learning individual patterns of behavior is not a completely new problem. In the SPECTER context, however, it is significantly aggravated by the fact that there exists nothing like the "/complete plan library"/ that has typically been assumed in plan-recognition research. Instead, while there may be more or less fixed sets of high-level goals and "/recipes"/ (operational descriptions of how to achieve a goal by performing a set of actions) for certain application domains, the set of available basic actions itself may be dynamic. The action alternatives depend not only on *U*'s capabilities but also on the current situation: the place, the resources, and the objects present (and detected by *U* or *S*). Additionally, new types of smart objects might appear, advertising new, previously unknown functionalities that have to be both integrated into *U*'s planning and decision making and taken into account by *S* when suggesting or performing actions on behalf of *U*. The ability to deal with this kind of dynamics on the level of both primitive actions and more abstract concepts will also be an important prerequisite for the extension of SPECTER's applicability to new domains.

Besides providing the basis for *U*'s decision making, the personal journal also serves as an *episodic memory*, i.e., a complex index to situations, persons, information, and objects. The aim is for SPECTER to be able to answer situated queries like "/What document did I read yesterday when my boss entered my office?"/ or "/Who was that guy we had a hard argument with at the last project meeting?'

Information provided by other users (e.g., via their own SPECTER systems) will be a further source of evidence for SPECTER's learning. The integration of such information must take into account differences among users. For example, a person who appears very friendly to user U_1 might be completely unappealing to U_2. When data and inferred assumptions about this person are transferred from U_1 to U_2, these differences have to be taken into account; in the simplest case, the system's confidence in such assumptions will simply be lowered. Another approach might be to restrict the exchange of information to pairs of users who are sufficiently similar. Here, a number of problems analogous to problems in the area of case-based reasoning will have to be tackled.

Finally, contradictory inferences might be made by a single SPECTER, if *U*'s behavior in certain situations changes in unforeseeable ways. The predictive models derived should provide hints as to which of a number of alternatives offered to *U* might be the best (for example, most probable, most likely to be successful, or most reliable). That is, qualitative or quantitative quality measures are necessary that allow the decision-making component—and also the user—to assess the quality of a recommendation and *S*'s confidence in it.

4.3 Augmentation of Decision Making and Effecting

SPECTER supports its user during her performance of tasks, serving as communication middleware for the cooperative interaction with intelligent interfaces in *U*'s environment. That is, SPECTER aims to achieve more efficient support and enhanced human interactions by connecting automatically to smart objects and by unobtrusively accessing the public profiles of persons who share a common task-specific context.

SPECTER is an always-active transceiver for the situation-dependent processing of sensory information: It receives information from various sources through its extended perception; and after analyzing these signals, it transmits the relevant and privacy-preserving parts of its user model to the environment in order to achieve useful adaptations and reactions. In this context, it considers the user's ongoing and scheduled tasks and the profile information available about the receiver. Another prerequisite for a SPECTER's active processing is access to its user's current goals and needs and to the current behavioral policies that govern its proactive behavior.

These are either learned from records stored in the personal journal or specified in a reflection phase when the user explores and adapts her personal journal and user model. These policies define the cases in which SPECTER is allowed to act autonomously and how SPECTER should interact with its user—for example, when it detects opportunities or would like to notify its user about important facts. SPECTER should provide concise explanations for its proactive behavior and the advice it gives. Its actions and the user's feedback are registered in the personal journal.

Since a SPECTER system will have to be secure against deliberate misuse, it will require a platform for privacy preferences (for example, like P3P, which is used in web applications), extended by trust technologies and methods for secure authentication, perhaps via biometrics. For example, the user can identify herself to her SPECTER by using a fingerprint identification chip integrated in the PDA. Then, the SPECTER becomes her personal appliance and acts on behalf of her. But as will be discussed below, even with the best technical security solution, *U* will have to be given opportunities to control the actions performed by SPECTER.

4.4 Reflection and Introspection

Whether a system like SPECTER will be appreciated by its users is in part a question of understanding and trust. It is therefore indispensable to provide an appropriate means for the user to learn about the records and models that the system has built up about her. Techniques such as textual, tabular, or graphical presentation seem unlikely to be adequate in themselves: First, the amount of data captured by SPECTER and the complexity of its user model will make exhaustive or unguided inspection infeasible. Second, it can seem unnatural or inappropriate for a machine to comment to a human about the human's (sometimes emotional) reactions to events and to actions of others.

So on the one hand, processes of reflection and introspection need to be investigated that allow the user to access and adjust SPECTER's records and models. But at the same time, factors like the manifestation of empathy, emotional argumentation, and the system's apparent sincerity need to be addressed. The emerging techniques will let SPECTER present observed emotional and behavior patterns in an affective way, taking into account an evolving longer-term user-SPECTER relationship. In turn, the user will be allowed to annotate past situations in terms of her own emotional responses, supplementing and perhaps correcting the automatically generated records of affective states.

Our research on this aspect of SPECTER will take into account the fact that the introduction of anthropomorphic elements into human-computer dialog does not automatically enhance the user's experience. In particular, a system like SPECTER will never be able successfully to perform the functions of a trained counselor, a priest, or an empathetic spouse. The goal is to find a satisfactory solution that lies between (a) the purely mechanical presentation of emotionally charged information and (b) human-like forms of interaction that raise expectations which cannot be fulfilled.

5 Personal Journal

As the SPECTER system accompanies the user as an unobtrusive, omnipresent spectator or consultant, it perceives and collects data of various kinds to build up a history of the different contexts the user goes through during her day.

The basic structure of the personal journal is inspired by the FORGET-ME-NOT system ([Lamming and Flynn, 1994]). It consists of a list of entries encoding past actions of the user, locations he went to, mobile services he used (or did not use), etc.

In general, we can distinguish two (not mutually exclusive) reasons why a context-aware system *S* may want to capture information about the user's current context and activities:

1. So that *S* take the information into account immediately for decision making.
2. So that *S* store the information for later analysis and/or querying.

Much like FORGET-ME-NOT, one of the main goals of SPECTER's personal journal is to serve as an episodic memory, i.e., a complex index to situations, persons, information, and objects. But SPECTER goes beyond the capabilites of FORGET-ME-NOT in that it not only presents a browsable personal history, but also allows for more complex retrieval. The aim is for SPECTER to be able to answer situated queries like "What document did I read yesterday when my boss entered my office?" or "Who was that guy we had a hard argument with at the last project meeting?"

Of course, that requires a deeper understanding of the journal entry than just a low-level sensor signal representation. The system integrates the different types of sensor data into an abstract form that aligns closer to *U*'s understanding of her own actions. In order to do so, SPECTER must be able to cope with heterogeneous input data that already come in at different levels of abstraction. For instance, a smart sensor in a store might exchange product information with specter by transfering subgraphs of a RDF ontology while a location sensor only transmits uninterpreted GPS coordinates. Thus SPECTER needs to find ways to process and synchronize qualitatively different levels of abstraction.

Another new aspect is that the system uses the personal journal together with the user model to reflect about *U*'s goals and behaves proactively to support her on them. An interpretation is built up from the journal data which is then evaluated with respect to *U*'s assumed current goals. This general approach arises the question to what extent *S* should interpret incoming data (e.g., about actions) while the information is still coming in? Some interpretation during input is certainly desirable, as often there is active perception required: *S* needs to redirect its sensors, send out queries, etc., in order to get the necessary information for an interpretation. For example, if SPECTER perceives information that a store has special offers on certain detergents, it may enquire whether this is also the case for the user's favorite brand. Also, *S* might want to take actions right away which requires immediate interpretation as well.

But still, it is not sufficient to do all interpretation at the time the data are perceived, dropping the raw data streams in favor of only storing the abstracted representations. It is possible that some additional interpretation which could not have been anticipated may become necessary at a later point ("Where could I have lost my wallet? "). This new interpretation is very likely to need access to the original (low-level) sensor data and not only to stored abstractions, because details that seem unimportant to one interpretation may well be of interest to another one.

Therefore, one of the most challenging question in that area is how *S* chooses the data to store permanently. The amount of sensor signals that the SPECTER system perceives in daily use can easily assumed to reach terabyte dimensions rather quickly. Although research on experimental systems like e.g. MYLIFEBITS ([Gemmell *et al.*, 2002]) suggests that even very large sizes of data will be effectively administrable in the near future, it does not seem very appealing to store just about everything. Firstly, since it can not be expected that these amounts of data can be stored only on the handheld device, the system would create vast network traffic just to transfer data to a server-sided database. Secondly, algorithms that work on those data, are forced to deal with more noise around the actual information of interest. Although this may not be a problem in offline computation, it certainly constrains *S*'s capabilities for immediate decision support.

6 User Model

As in practically all ubiquitous computing scenarios, the information about *U*, stored in *S*'s user model, need to be administered carefully. On the one hand, the user's privacy must not be violated, but on the other hand, there are situations where personal information has to be communicated to third party components. For example, a mobile service may charge a small fee for its use, so the user would like her SPECTER to give out credit card information to this service without asking for permission every single time. Of course, the user should always be in control to which entity personal information is transferred.

It has been pointed out (e.g., by [Jameson, 2002]) that privacy violations can occur even if there is no technical security leak or intentional attempt to gain unauthorized access to data. In any case where SPECTER's behavior is visible to persons other than the user, these other persons may be able to infer information about the user that is stored in SPECTER and that gave rise to *S*'s behavior. For example, if SPECTER spontaneously turns on a TV to show *U* a wrestling match, or audibly recommends a nearby bar, others may indirectly infer previous behaviors of *U* that indirectly gave rise to these system behaviors.

One important conclusion is simply that no technical solution can fully guarantee users' privacy. Even aside from cases where the solution fails because of technical defects and/or maliciousness of some of the parties involved, there will always be a tradeoff between measures taken to protect privacy and the benefit that users can derive from the system. Results of [Ackerman *et al.*, 1999] in connection with e-commerce sites suggest that users will differ strongly in their willingness to provide particular types of information—and indeed in their overall attitudes toward privacy. Several conclusions of the authors—which seem plausible for systems like SPECTER as well, are the following:

- Users must be given opportunities to control various aspects of the privacy policies that are applied to them.
- Whereas some users will be satisfied with fairly simple, global preferences, others will wish to specify different preferences in different situations.
  This issue of the situation-specificity of preferences seems likely to be even more important with a mobile system like SPECTER, which is used in a greater variety of contexts that, say, a traditional web-based e-commerce system.
- The goal of allowing users to customize a privacy policy raises significant challenges for user interface design—just as the tracking of complex and changing preferences is a challenge for other types of preference-dependent systems as well, such as product recommendation systems (see, e.g., [Jameson *et al.*, 2002]).

Again, this problem is further exacerbated by the limited communication bandwidth of mobile devices and by the fact that users can often not devote full attention to the operation of the device.

7 Conclusion and Future Work

We have given an overview of our project SPECTER, that aims to bear on some of the still open challenges in the field of ubiquitous computing. We have pointed out a few important issues that generally need consideration in mobile computing, like

1. the ability to cope with different levels of sensor data and how to synchronize, abstract and interpret them if they are to be presented directly to the user as in SPECTER's personal journal.
2. ways how to use context information to support the user's episodic memory as well as provide immediate decision support.
3. privacy concerns that do not only stem from technical deliberations, but arise from the user's mere interaction with the system.

Due to the very early state of our research, this essay raises more questions than it answers. Still, we hope that the approaches mentioned herein provide a valuable share to other researchers.

At the current time, our group is working on the implementation of a very first prototype. We are optimistic that we will be able to use it as a testbed for future research. The prototype is going to realize a subset of shopping scenario where environmental sensor data are still completely simulated. Also the handheld client device will at first be simulated, but the implementation will already take the portation to a PDA into account. The next step would then be to substitute all other simulated parts of the system with real-world counterparts.

The concrete design of both the personal journal and the user model as the key components of SPECTER is a task that we are working on in parallel to the prototype realization. Thereby, the challenges outlined in this essay need to be addressed notably.

References

[Ackerman *et al.*, 1999] M. S. Ackerman, L. F. Cranor, and J. Reagle. Privacy in e-commerce: Examining user scenarios and privacy preferences. In *ACM Conference on Electronic Commerce*, pages 1–8, 1999.

[Albrecht *et al.*, 1997] D. W. Albrecht, I. Zukerman, A. E. Nicholson, and A. Bud. Towards a bayesian model for keyhole plan recognition in large domains. In A. Jameson, C. Paris, and C. Tasso, editors, *User modeling: Proceedings of the Sixth International Conference, UM97*, pages 365–376, Vienna, 1997. Springer Wien New York.

[Bauer, 1998a] Mathias Bauer. Acquisition of abstract plan descriptions for plan recognition. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 936–941, Madison, WI, 1998.

[Bauer, 1998b] Mathias Bauer. Towards the automatic acquisition of plan libraries. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence*, pages 484–488, 1998.

[Bauer, 1999] Mathias Bauer. From interaction data to plan libraries: A clustering approach. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 962–967, San Francisco, CA, 1999. Morgan Kaufmann.

[Byun and Cheverst, 2002] H. E. Byun and K. Cheverst. Harnessing context to support proactive behaviours. In *Proceedings of ECAI Workshop on AI in Mobile Systems*, 2002.

[Chandrasekaran and Josephson, 2000] B. Chandrasekaran and J. R. Josephson. Function in device representation. *Engineering with Computers: Special Issue on Computer Aided Engineering*, 16:162–177, 2000.

[Chandrasekaran, 1994] B. Chandrasekaran. Functional representation and causal processes. *Advances in Computers*, 38:73–143, 1994.

[Davison and Hirsh, 1998] B. Davison and H. Hirsh. Probabilistic online action prediction. In *Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments*, 1998.

[DeVaul *et al.*, 2000] R. W. DeVaul, B. Clarkson, and A. S. Pentland. The memory glasses: Towards a wearable, context aware, situation-appropriate reminder system. In *In Proceedings of CHI Workshop on Situated Interaction in Ubiquitous Computing*, 2000.

[Dey and Abowd, 2000] A. K. Dey and G. D. Abowd. Cybreminder: A context-aware system for supporting reminders. In *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing*, pages 172–186, 2000.

[Garland *et al.*, 2000] A. Garland, N. Lesh, C. Rich, and C. L. Sidner. Learning task models for collagen. Technical Report TR2000-36, Mitsubishi Electric Research Laboratories, 2000.

[Garland *et al.*, 2001] A. Garland, K. Ryall, and C. Rich. Learning hierarchical task models by defining and refining examples. In *In Proceedings of the First International Conference on Knwoledge Capture*, 2001.

[Gemmell *et al.*, 2002] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. Mylifebits: Fulfilling the memex vision. In *ACM Multimedia 2002*, Juan Les Pins, France, 2002.

[Hanssens *et al.*, 2002] N. Hanssens, A. Kulkarni, R. Tuchinda, and T. Horton. Building agent-based intelligent workspaces. In *Proceedings of Workshop on Agents for Business Automation*, 2002.

[Horvitz *et al.*, 1998] Eric Horvitz, Jack Breese, David Heckermann, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In G. F. Cooper and S. Moral, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference*, pages 256–265, San Francisco, 1998. Morgan Kaufmann.

[Horvitz *et al.*, 1999] Eric Horvitz, Andy Jacobs, and David Hovel. Attention-sensitive alerting. In K. B. Laskey and H. Prade, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Fifteenth Conference*, pages 305–313, San Francisco, 1999. Morgan Kaufmann.

[Horvitz *et al.*, 2002] Eric Horvitz, Paul Koch, Carl M. Kadie, and Andy Jacobs. Coordinate: Probabilistic forecasting of presence and availability. In A. Darwiche and N. Friedman, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference*, pages 224–233, San Francisco, 2002. Morgan Kaufmann.

[Jameson *et al.*, 2002] Anthony Jameson, J. A. Konstan, and J. Riedl. Ai techniques for personalized recommendation (tutorial). In *18th National Conference on Artificial Intelligence*, 2002.

[Jameson, 2002] Anthony Jameson. Designing systems that adapt to their users (tutorial). In *CHI 2002 Conference on Human Factors in Computing Systems*, 2002.

[Koychev, 2002] I. Koychev. Tracking changing user interests through prior-learning of context. In P. De Bra, P. Brusilovsky, and R. Conejo, editors, *Adaptive hypermedia and adaptive web-based systems: Proceedings of AH 2002*, pages 223–232, Berlin, 2002. Springer.

[Lamming and Flynn, 1994] Mik Lamming and Mike Flynn. "forget-me-not": Intimate computing in support of human memory. In *FRIEND21, the 1994 International Symposium on Next Generation Human Interface*, Meguro Gajoen, Japan, 1994.

[Sakamoto *et al.*, 2001] Ryuuki Sakamoto, Keiko Nakao, Yasuyuki Sumi, and Kenji Mase. Comicdiary: Representing individual experiences in comics style. In *the Twenty-Fifth International Conference on Computer Graphics and Interactive Techniques*, page 158, 2001.

[Salber *et al.*, 1999] D. Salber, A. K. Dey, and G. D. Abowd. The context toolkit: Aiding the development of context-enabled applications. In M. J. Tauber, V. Bellotti, R. Jeffries, J. D. Mackinlay, and J. Nielsen, editors, *Human factors in computing systems: CHI '96 conference proceedings*, pages 434–441, New York, 1999. ACM.

[Shafer *et al.*, 2001] S. A. Shafer, B. Brumitt, and J. J. Cadiz. Interaction issues in context-aware intelligent environments. *Human Computer Interaction*, 16:363–378, 2001.

[Sumi and Mase, 2001] Yasuyuki Sumi and Kenji Mase. Agentsalon: Facilitating face-to-face knowledge exchange through conversations among personal agents. In *the Fifth International Conference on Autonomous Agents*, pages 393–400, 2001.

[Widmer, 1997] G. Widmer. Tracking context changes through meta-learning. *Machine Learning*, 27(3):259–286, 1997.

[Yoshida and Motoda, 1995] K. Yoshida and H. Motoda. Clip: Concept learning from inference patterns. *Applied Artificial Intelligence*, 75:63–92, 1995.

[Yoshida *et al.*, 1994] K. Yoshida, H. Motoda, and N. Indurkhya. Graph-based induction as a unified learning framework. *Applied Artificial Intelligence*, 4:297–316, 1994.