Towards Goal Elicitation by User Observation

Sven Schwarz and Thomas R. Roth-Berghofer
German Research Center for Artificial Intelligence DFKI GmbH
Erwin-Schrödinger-Str. 57, 67663 Kaiserslautern
{Sven.Schwarz | Thomas.Roth-Berghofer}@dfki.de

Abstract

Knowledge workers put a lot of effort during every day's work in structuring their own information. This is often done with the help of email tools, file directories, or bookmarks. The recently started research project EPOS[1] investigates ways to assist users in providing adequate and task-specific support. The envisioned system will observe the user's work as well as his ways of information handling and automatically learn and identify his goals, intentions, structures, ontologies, and work processes. Towards the user, a sophisticated knowledge workspace shall act as an adaptive assistant proposing follow-up working steps and providing (how-to) information as well as relevant documents. In order to do so, the assistant needs to know about the user's current *context*. One part of that context is the user's current goal. The user goals are represented by *task concepts* which are concepts in an ontology. We propose to use Case-Based Reasoning (CBR) techniques to predict the user's goal(s) from a sequence of his workspace (inter-) actions needed to achieve this goal.

1 Assisting Knowledge-Intensive Work

Knowledge workers [Davenport and Prusak, 1998] put a lot of effort during everyday's work in structuring their own information. This is often done with the help of email tools, file directories, or bookmarks, but the way this is done depends highly on the individual manner of working. Contrary to that are the knowledge management goals of a company: global collection, structuring and distribution of knowledge. Hence, there is a gap between the global benefits for the organization and the personal benefits for the individual knowledge worker. While the organization asks for universally applicable, standardized and persistent structures, processes, etc. to achieve and maintain universally accessible information archives, the individual knowledge worker requests his common, individualized structures and flexibility in processes and work organization in order to reach optimal support for his activities. Therefore, the recently started research project EPOS [Dengel *et al.*, 2002] investigates a bottom-up evolutionary approach to narrow this gap.

The individual knowledge workspace is realized mainly by a set of applications running on the user's personal computer. In order to provide adequate and task-specific support to the human, the system will observe the work and the users' ways of information handling and automatically learn and identify his goals, intentions, structures, ontologies, and work processes. Towards the user, the knowledge workspace shall act as an adaptive, context-aware assistant proposing follow-up working steps and providing (how-to) information, as well as, relevant documents. Therefore, the assistant needs to know about the user's current *context* which has, according to [Maus, 2001; Turner *et al.*, 2001], many components, dimensions, or *contextual aspects*.

In order to be really useful for the user, the assistant has to be as unintrusive as possible, i.e., most of the context should be gathered from observing the user instead of depending on being fed explicitly by the user. EPOS will observe the user at his daily work on his PC and elicit as much user context as possible out of this observation. This context information will then be used for context-sensitive assistance functionality.

The paper is organized in the following way: In section 2 we introduce a hierarchy of four abstraction levels of user activities, followed by an example that guides through the four levels in section 3. Then, we describe the knowledge sources we deem important from a more technical view in section 4 before we sketch our proposed goal elicitation approach using Case-Based Reasoning techniques in section 5. We close the paper with a brief discussion of open issues and an outlook on further work of the EPOS project team.

2 Four Abstraction Levels of User Activity

Figure 1 depicts four different levels of user activity: The first level, called **Workspace Level**, represents the operating system and the applications that provide access to files, objects and information structures. Observation at this level results in workspace events such as various mouse clicks, entering of some text, or starting and handling applications.

The user's momentary intentions, expressed by his *user actions*, are independent of the currently used workspace. The **User Action Level**, thus, contains such user actions as *create new text document* or *revise document*, rather than atomic mouse-clicks or actions like *start text editor* or *activate File-new menu*. Those user actions will be inferred from a series of workspace events described before. We do not address this in this paper. This is another part of research of the EPOS project.

While the user tries to solve his problems with the OS and some applications, he always has some higher medium-

[1] Evolving Personal to Organizational knowledge Spaces, `http://www.dfki.de/epos`

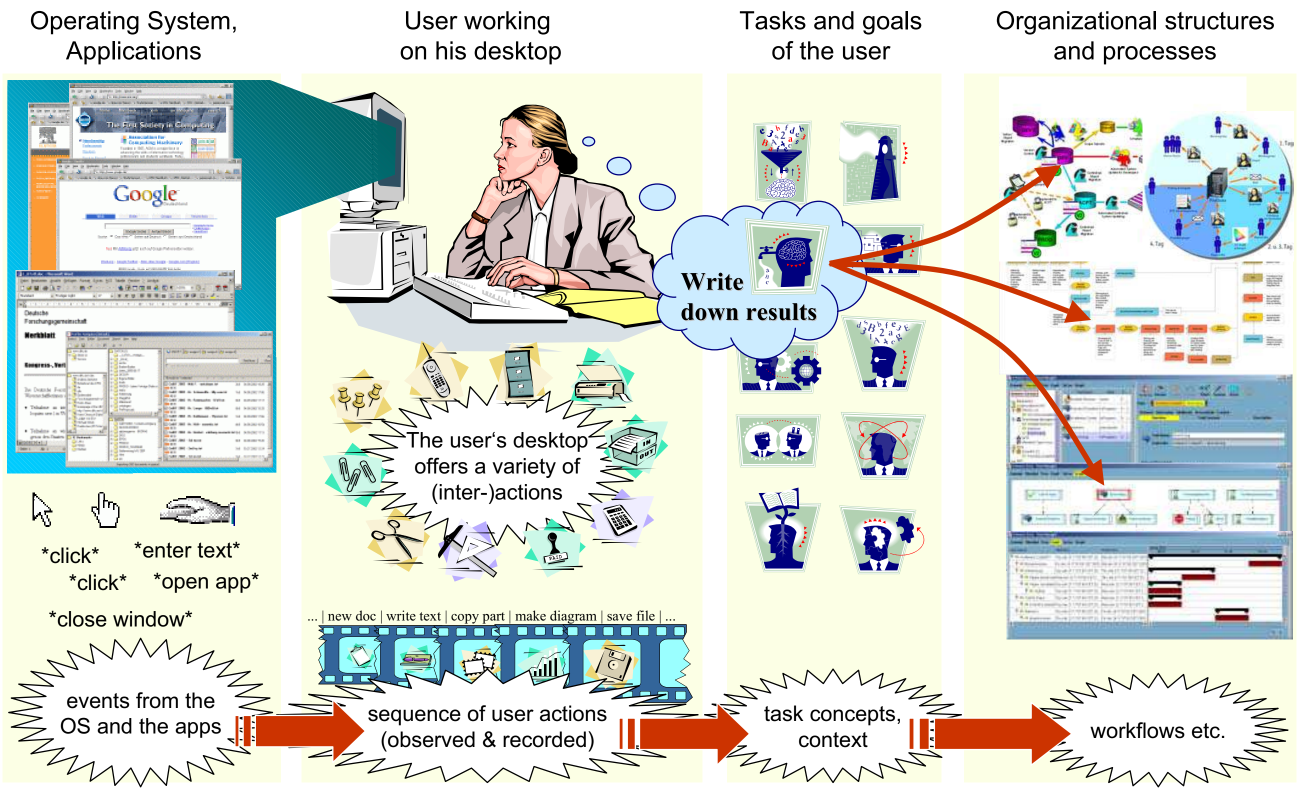

Figure 1: The user observation architecture handling different abstractions of user interactions.

term goal in mind such as *write down results* or *write project proposal*. Those user goals are captured in the **Task Concept Level** and are represented by *task concepts* which are concepts in an ontology about such user goals. EPOS will elicit the user's goal(s) from a sequence of the user actions needed to be carried out to achieve this goal. This paper will focus on a mechanism for this elicitation problem.

Knowing about the user's goal(s), EPOS can provide goal specific support to the user such as relevant documents / (how-to) information. Furthermore, if we know about goal specific (generic) information-needs, we can fill the user's information-need gap by presenting him respective documents.

And, last but not least, the **Process Level** connects to the organizational structures processes which might be explicitly modelled, e. g., with a business process modeling tool or enacted by a workflow management system (WfMS). If there is such a WfMS available, we can connect / assign the user to running workflows. Workflows can be semantically described [Schwarz, 2003] using the same set of task concepts as we're eliciting from the user's behavior. So, we can use the task concepts to identify the workflow tasks the user is (or seems to be) currently working on. That way we can use and offer workflow knowledge indirectly, i. e., without *direct* interaction with a WfMS.

3 Usage Example

An example usage scenario shows the different layers of abstractions, as well as, the envisioned user assistance.

The first thing available to the observation module of the EPOS system is the user's workspace activity resulting in a series of workspace actions / events like the following:

1. start text editor
2. open a text file
3. scroll around a bit
4. print out document
5. close text editor

The system tries to elicit direct user intentions (*user actions* out of these native workspace events. The system can, however, not be certain about elicited user actions. Therefore, we have to cope, generally, with a multitude of more or less probable user actions:

- skim over this document (most probable)
- print out this document (still probable)
- (physically) archive this document (possible)

This user action elicitation is a continuous process, i. e., the user will continue to use the workspace and, as such, produce further workspace events, e. g.,

1. open web browser
2. go to URL of citeseer
3. enter some text
4. press search button
5. ...

The system will gather those workspace events and, again, try to elicit respective user action(s) out of it, e. g.,

- search for document(s) (very probable)
- search for author(s) (still probable)

For the sake of simplicity, we assume only to consider the most probable user action and leave out the other ones. As time passes, we then get a chronologically ordered sequence of most probable user actions, like this one:

$$\begin{bmatrix} \text{action} = \text{skim over doc} \\ \text{app} = \text{text editor} \\ \text{doc} =< \text{doc-id} > \end{bmatrix} \rightarrow \begin{bmatrix} \text{action} = \text{search for docs} \\ \text{app} = \text{web browser} \\ \text{search engine} = \text{google} \\ \text{terms} = \{\text{CBR, context}\} \end{bmatrix}$$

Using the mechanisms described in this paper, we will use this *user action sequence* to elicit the user's goal (task concept). As before we have to cope with ambiguity as long as the user isn't helping the elicitation process by some feedback. We get a set of probable task concepts like the following:

- investigate related work (very probable)
- research state of the art (very probable)

Using the estimated task concepts we can use the semantic descriptions of workflows and work items to identify a potential assignment of the user to an active work item. We may receive the following hypotheses:

- user works (most probably) at work item *0306201*, a work item named "related work"
- follow-up work item in the same workflow is *0306202*, a work item named "related research projects"

4 Knowledge sources and their structure

Figure 2 shows a more detailed, technical, and complete view of our envisioned user observation and context elicitation framework. As indicated by bigger arrows, this paper focuses on the identification of the user's goals (task concepts) utilizing recently observed user actions. This mapping will be done using Case-Based Reasoning (CBR).

To be more precise, consider a vector representing a sequence of the most recent user action objects: $(ua_{t-n}, \ldots, ua_{t-1}, ua_t)$. The vector describes past to present user actions. Its length, and hence the view into the past, must be restricted to a capable length, of course. Determining the length of the vector and the selection of "interesting" combinations of user actions will be the topic of another paper.

A user action object stores different aspects of information about and around the mere action invoked by the user such as

- invoked action (e. g., *enter text*),
- used application (e. g., some *text editor*),
- *text context*: text around the user's text cursor (or alternatively the mouse cursor), e. g., four sentences before and 1 sentence after the cursor or surrounding paragraphs,
- *domain* of the text context (e. g., *workflow management, knowledge intensive work, user observation*), and
- document type of the touched document (e. g., *project proposal*).

Hence, the user action objects and, especially, the user action sequences are complex structured objects.

In parallel to the observed user actions, there's a *workspace model* (cf. Figure 3) describing objects available to the user plus possible operations / actions on these objects. This provides additional domain knowledge for the mapping mechanism.

Moreover there are several ontologies which support the elicitation even further:

- relationships between domain concepts (e. g., *workflow management is-related-to business process modeling*)
- relationships between document types (e. g., *project proposal is-a proposal, proposal is-a paper*)

5 Approach

The goal elicitation problem described above, can easily be formulated as a Case-Based Reasoning classification task[2], i. e., determine the user's most probable goals (task concepts) for a given sequence of user actions.

Our application domain is a weak theory domain where the important causal interactions are not well understood (and not easily to grasp, anyway). There is no correctness criterion available which can determine that a given sequence of user actions belongs to a certain task concept. At this point, relevance feedback will have to be regarded. The relation between a sequence of user actions and task concepts is user related and may change for a given user over time.

In terms of the SIAM methodology for building and maintaining CBR applications [Roth-Berghofer, 2003], most of the Setup phase has been completed: the project objectives are set, a project team has been selected, etc. (Of course, one has to keep in mind that SIAM was developed in the context of *commercial* CBR applications. EPOS, on the one hand is a research project, and, on the other hand, its goal is not mainly to build a CBR system.)

But, nevertheless, let us follow the SIAM path further. The next phase is the Initialization phase. This phase requires to acquire knowledge to create cases, determine an initial domain model and to build an initial case base.

The cases in this domain are classical cases of rule type [Richter, 1997]. They describe a problem solving episode consisting of a problem specification, i. e., a sequence of user actions, and a solution, i. e., a concept of the corresponding task ontology. Of course, our underlying assumption is that similar user actions indicate similar goals, at least in certain contexts; an assumption that has to be validated through further research and experiments.

We currently represent a case as a vector of attribute value pairs as described above. The cases are records of past events, and, thus, episodic in nature [Watson, 1999]. In the course of the knowledge acquisition for the cases, we will develop the initial domain model. Part of the vocabulary knowledge container [Richter, 1995] is already given by an ontology of user actions and the task concept ontology. But this has to be developed in more detail.

Another challenge will be the development of appropriate similarity measures. Questions to answer are, e. g., how

[2] cf. [Lenz *et al.*, 1998] for a description of CBR application types

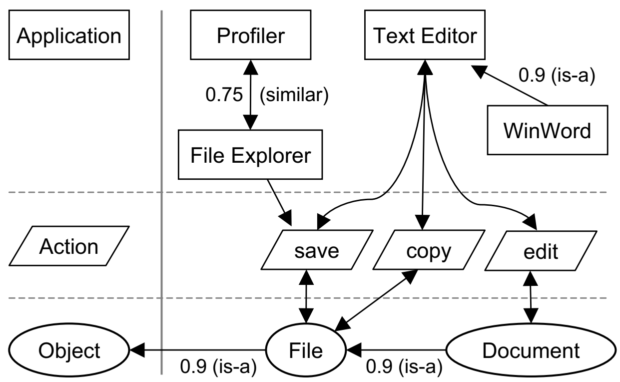

Figure 3: Extract of a simple workspace model of interrelating applications, actions and objects of a user's desktop.

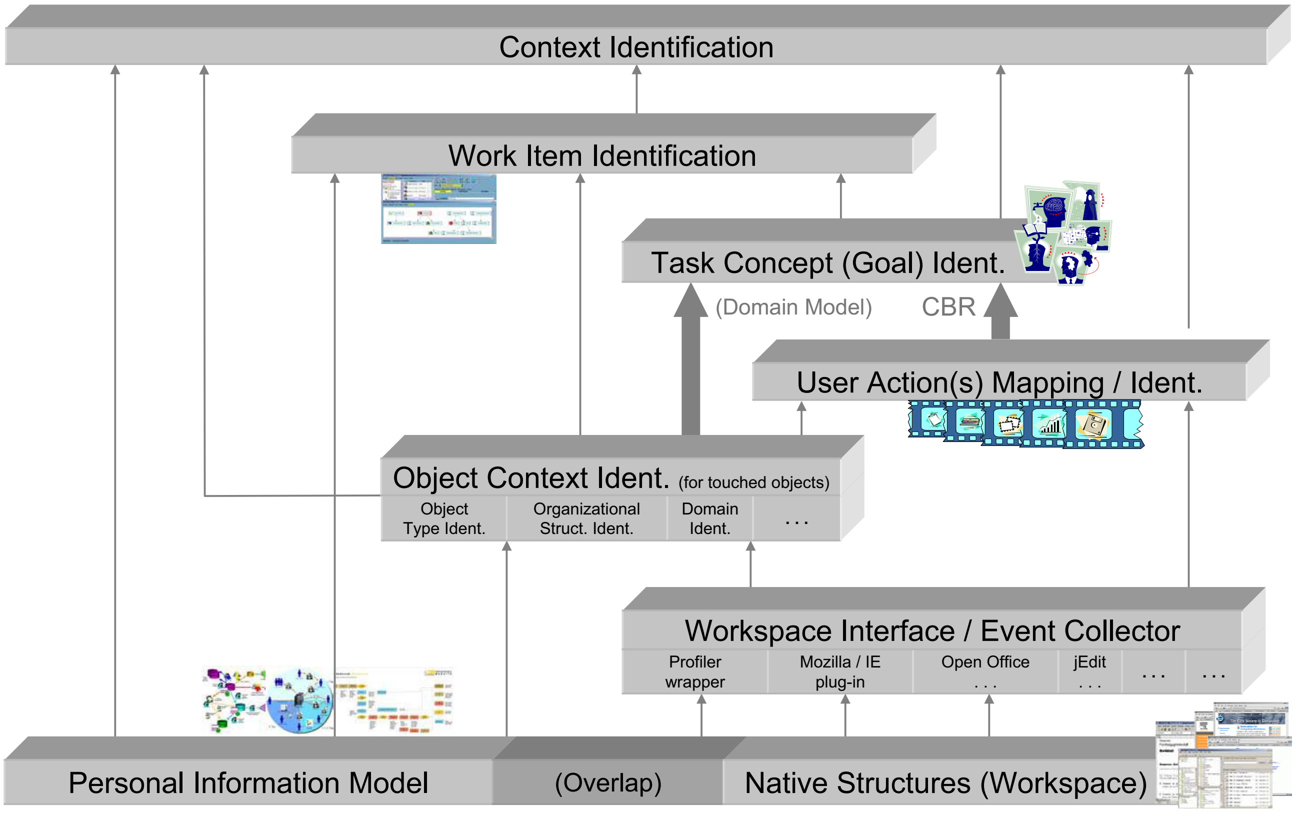

Figure 2: The user observation and context elicitation components and their interplay.

many elements should a user action sequence contain to make up a case or how important is the time order of the user action sequences. An in depth analysis user actions acquired in experiments is intended to form a basis for the similarity measures.

The experiments mentioned will also help to build an initial case base. Appropriate cases have to be built from user action sequences and corresponding task concepts as their "solutions".

We do not need to take care of adaptation knowledge. This knowledge container is empty because the problem is a classification problem that does not require solution transformations per se.

6 Things for Discussion and Future Work

As already mentioned, our research project has just started and we tried to sketch the problem area together with a potential solution for some specific part, i. e., the mapping from the user action level to the user task level. Apart from the goal elicitation problem, there are further issues to regard. For example, the context elicitation is *not* trivial at all, because we will have to handle *multiple/parallel*, *fuzzy/ambiguous* contexts:

- As the user's daily work typically requires concurrent work on multiple and different (distinct) processes, the context elicitation has to cope with a multitude of parallel streams of work and goals.
- The more automatically the context elicitation acts, the more we have to cope with uncertainty and/or fuzzyness in the context.

Then, what about handling multiple users / desktops / workspace models etc.? The current approach is a stand-alone (single person) only solution. By now, we focus first on the case of one single person, one desktop PC, one task concept ontology [Schwarz, 2003] and one trained case base for task concept identification (i. e. user goal elicitation). Storing user action sequences under relevant (identified) task concepts leads to an instance based definition of task concepts. Hence, as long as two persons use the same workspace model (domain model) and the same set of user actions, we can negotiate about task concepts via stored sets of user action sequences.

Furthermore, context elicitation won't work without user interaction especially not at the beginning of training and ontology construction. How will such user interaction look like?

- in combination with visualization (especially context visualization)
- in form of feedback ("yes!" / "no!") buttons
- important: balance user benefits against user disturbance!

We look forward to using the workshop as a platform for discussing problems emerging wrt. our envisioned goal elicitation.

Acknowledgments

The research project EPOS is funded by the German Federal Ministry for Education, Science, Research and Technology (`bmb+f`) under contract number `01 IW C01`.

References

[Akman *et al.*, 2001] Varol Akman, Paolo Bouquet, Richmond H. Thomason, and Roger A. Young, editors. *Modeling and Using Context, Third International and Interdisciplinary Conference, CONTEXT, 2001, Dundee,* *UK, July 27-30, 2001, Proceedings*, volume 2116 of *Lecture Notes in Computer Science*. Springer, 2001.

[Davenport and Prusak, 1998] Thomas H. Davenport and Lawrence Prusak. *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston, Massachusetts, 1998.

[Dengel *et al.*, 2002] Andreas Dengel, Andreas Abecker, Jan-Thies Bähr, Ansgar Bernardi, Peter Dannenmann, Ludger Van Elst, Stefan Klink, Heiko Maus, Sven Schwarz, and Michael Sintek. Evolving personal to organizational knowledge spaces. Project Proposal, DFKI GmbH Kaiserslautern, 2002. `http://www.dfki.de/epos/` [Last access: 2003-07-25].

[Lenz *et al.*, 1998] Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, and Stefan Wess, editors. *Case-Based Reasoning Technology: From Foundations to Applications*. Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 1998.

[Maus, 2001] Heiko Maus. Workflow Context as a Means for Intelligent Information Support. In Akman et al. [2001].

[Richter, 1995] Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995. `http://wwwagr.informatik.uni-kl.de/~lsa/CBR/Richtericcbr95remarks.html` [Last access: 2002-10-18].

[Richter, 1997] Michael M. Richter. Generalized planning and information retrieval. Technical report, University of Kaiserslautern, Artificial Intelligence – Knowledge-based Systems Group, 1997.

[Roth-Berghofer, 2003] Thomas R. Roth-Berghofer. *Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology*, volume 262 of *Dissertationen zur Künstlichen Intelligenz*. Akademische Verlagsgesellschaft Aka GmbH / IOS Press, Berlin, Germany, 2003.

[Schwarz, 2003] Sven Schwarz. Task-Konzepte: Struktur und Semantik für Workflows. In Ulrich Reimer, Andreas Abecker, Steffen Staab, and Gerd Stumme, editors, *WM2003: Professionelles Wissensmanagement – Erfahrungen und Visionen*, pages 351–356. GI, 2003.

[Turner *et al.*, 2001] Roy M. Turner, Elise H. Turner, Thomas A. Wagner, Thomas J. Wheeler, and Nancy E. Ogle. Using Explicit, A Priori Contextual Knowledge in an Intelligent Web Search Agent. In Akman et al. [2001], pages 343–352.

[Watson, 1999] Ian Watson. Survey of CBR application areas. 1999. Invited Talk at the 3rd International Conference on Case-Based Reasoning.