Situated learning in the work process

Reinhard Oppermann

Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
Reinhard.Oppermann@FIT.Fraunhofer.de

Abstract

Situated learning in an authentic process is the most natural kind of learning. If there are no restrictions for learners like media accessibility, trainer and co-learner availability, time and location constraints every learner would learn when s/he needs the respective knowledge or skill. But actually there are restrictions. During the history of media development there were accessibility limitations like price, form factor, usability in specific environments no matter whether the media were a scribble, a book, a slide, a video or a computer screen. The same is true for teachers and trainers who are not available every time and everywhere on demand but have to be organized for specific sessions for specific learner groups. Finally restrictions also exist for the cognitive learning process. During the learning process no isolated steps of knowledge acquisition take place but interlinked knowledge construction is based on existing patterns of knowledge extending the complexity of understanding and the range of competence in a given area. Learners continuously extend their knowledge by exploration and by experience.

The paper focuses on contributions to situated learning in the process of the acquisition, use and reuse of knowledge together with other partners in different roles like co-learners, instructors, pre- or post-processors of content. Annotation facilities for situated learning and relearning are introduced and experiences are reported. Beyond static annotations episode recordings and mobile collection facilities are presented for dynamic sequences of problems or solutions.

1 Introduction

Three aspects are essential for situated learning to be effective and efficient:

1. Embedding learning in a process
2. Embedding learning into the context of use
3. Supporting learning by equipment suitable for flexibility and mobility

Embedding learning in a process means that for the learner (or the learning community) the learning process is transparent in terms of learning steps, learning trials and errors, contributions of other actors (teachers, trainers, supervisors, peers), tools and methods used. If the learner can see and explore the learning process, if the learner can reuse parts of the own learning history in later situations, if the learner can create and receive information about the tools and methods used then the learner can profit from his/her own learning process. We all know such support methods for learning and work where people stick post-its to their environment to remind themselves to critical events and methods. In electronic learning environments such ways of memory sticks are more easily and effectively possible and can coexist with the electronic working environment in a personalised way.

Embedding learning into the context of use helps to integrate the learning process into the working process. Many learning events occur during the work performance - we call it learning on demand. Learning on demand means that it takes place without a previous plan for learning, without a curriculum of learning, without explicit organisation for learning. Learning on demand in the context of work starts with a problem and ends with a solution in the given situation. Sometimes there is only one step between the awareness of a problem and the found solution. Sometimes a long process of exploration and consultation is needed to find the solution.

The integration of whole learning activity into the work process is not possible. You cannot start working on a zero level of competence. Apprenticeship approaches have shown that observation, supervised action and out fading of control for increasing autonomy are crucial steps in the process of competence development in praxis. Dependent on the complexity of the task basic or extended or even advanced competence is needed for working.

Over decades users of IT were accustomed to use stationary equipment. Since a couple of years mobile equipment complements this stationary infrastructure and frees the user from these ties. Mobile equipment like PDAs, mobile phones, tablet-PCs and in particular situations also wearable computing hardware also supports learning on the move. The innovative potential of learning on the move is not the opportunity to learn philosophy lessons at the bus stop. The use of free time slots for simple memorisation may be a possibility. But the essential innovation is the chance to use learning aids at the site of work, i.e., where the task is to be performed or where the problem is to be solved.

In this paper we will show concepts of and empirical evidence for situated learning in the context of work in two areas of application. First we will show how employees can profit from reflective annotation and recording tools for stepwise learning in the work process. Second we will show at least the principle idea of how students can profit from remote documentation and communication tools for collaborative learning and how service personnel can profit from mobile instruction and communication tools for situated problem solving.

2 Theoretical background and empirical findings for situated learning

Employees in modern work settings use more and more technologies and work procedures on the fly, i.e. without formal introduction and training before the job. Most knowledge acquisition takes place on the job (Paul 1995; Dugas & Batschkus et al. 1997; Dutke & Schönpflug, 1987). Many software products include tutorials and online help for supporting users at the work place so that users can explore the system in use and try out functions to reach their goals. This learning on demand increases the motivation for and the application of learning: "...people learn best when engrossed in the topic, motivated to seek out new knowledge and skills because they need them in order to solve the problem at hand" (Norman and Spohrer 1996). Knowledge not used in practice is difficult to retrieve and difficult to use when actually needed (Schank & Kass 1996). 'Guided exploration' facility as proposed to support this kind of learning (Carroll 1990) is just a technical answer to well known pedagogical principles of 'discovery learning' (Williams & Farkas 1992).

Help from the technical system is limited to information about system functionality and to well-known notorious problem situations (Fox, Grunst et al. 1994). The support users get is restricted to the information that experts have brought into the system in advance. There are only few examples for growing support based on questions of users and answers of consultants: 'Answer Garden', see (Ackerman & Malone 1990).

Beside technical support like tutorials, online help, guided exploration, being produced by experts for the user lack the individual perspective of the current context of use. They are often criticized for its inefficiency and ineffectiveness and its authoritarian nature (Williams & Farkas 1992).

As an alternative or at least as a complement personally consulting human experts is a way many users prefer. So, while the learning process may start with the user who helps him- or herself by exploration (trying things out) but then the user asks for help by consulting a competent colleague ('poweruser') in face-to-face interaction or by consulting an expert by telephone or remote diagnose.

Learning supported by computer help and documentation without social support is not appreciated by many users. Users tend to prefer constructive and co-operative communication between humans with complementary types of knowledge and expertise but being familiar with the same tasks and the same working environment, speaking the same jargon (O´Malley 1986; Brockmann 1990; Horton 1990). Instead of separate individuals most work is organised in work communities where the members of the work group share knowledge and profit from synergies between types and areas of competencies: "End users make good use of other people in their social environments to help them solve their computing problems and to compensate for gaps in their own knowledge of computers" (Nardi 1993). Local experts can be enlarged by professionals with technical knowledge about the system in use but with less connection to the user community and the task at hand. The latter are less accessible for and less accepted by the users (Bannon 1986).

There will be a process of learning different areas of the application's functionality, in particular with occasional or 'discretionary users' (Santhanam & Wiedenbeck 1993). Communities of system users will emerge, in which individuals have different backgrounds of knowledge: substantial computer and substantial domain expertise distributed among different members of the community. The competence of the user groups together with the competence of professional system experts is the basis for their constructive interaction in problem solving.

User support by personal interaction is limited by the capacity and availability of human experts. Empirical studies show that users have problems with consultants and consultants have problems with their clients (Brezizinski 1987; Liechti 1988; Moning & Winkelmann 1993). Consultants are overloaded; their growing number is overcompensated by a yet increasing number of clients; members of the user service units show limited availability; consultants have to solve (in their eyes) trivial problems and are therefore not motivated to invest their time into repeated individual support.

This is the background of the development of annotation and recording facilities for the reflection and communication of problems and solutions in the context of use.

3 Reflective annotation and recording tools for stepwise learning

While working with an application and while solving problems and performing tasks the people try to memorise how the problem appeared and the solution was carried out to refer to the details in later similar situations or to describe them to peers or supporting experts. Two methods will be presented that support the memorization of such details integrated in the work process. The first, called contextualised annotations, are flexible containers for static content in terms of text, graphics or tables. The second method, called episode recordings, is a container for sequences of actions for the memorization of the dynamics of problems and solutions. Both methods are described below.

3.1 Contextualised annotations

By contextualised annotations users get a tool to stick an electronic post-it to any element of their working environment: a function, an icon, a label, an entry field etc. They can glue the post-it annotation by dragging a “needle with a note” (see figure 1) from the toolbar of the application and releasing the mouse button at the respective destination of the current element, where the annotation is needed.

Annotations can be created as an experience based individual help for later use of the individual. Annotations can also be created by peers or by special roles like consultants, administrators, power-users for the users to profit from peer or specialist expertise for their own productivity.

Annotations can be used to describe the logic and usage of a function, the pitfall of procedures, the conventions of the working group or company. Annotations can be entered as written text in a text box, as a graphic or table or as a spoken comment.

Annotations can be created as a personal or as a public note. The personal note is only open for the user him- or herself. The public note is open for the user community. The creator of the note can define the class of note during or after creating the note.

The following figure 1 shows an example of an annotation created for printing shop software.

The application contains the tool as an additional entry for creating notes wherever necessary to communicate an individual or community-open annotation. The tool is located at the upper right corner. It shows a yellow note with a needle to symbolize an attachable sheet of paper. At the destination of the note at the right side an icon is attached by the user showing the place, where the note can be created in the editing situation and can be read in the later consultation situation. After releasing the mouse button in the editing situation or after clicking on the icon in the consultation situation the editing window opens up to enter the comment: in the example shown in figure 1 a description of several classes of clients of the printing shop based on the turnover of the clients with the printing shop is shown to specify price conditions for the client. The note is created by the head of the department to communicate the client standards to the users who write offers for clients. Therefore the note in the example has been specified by the creator as public.

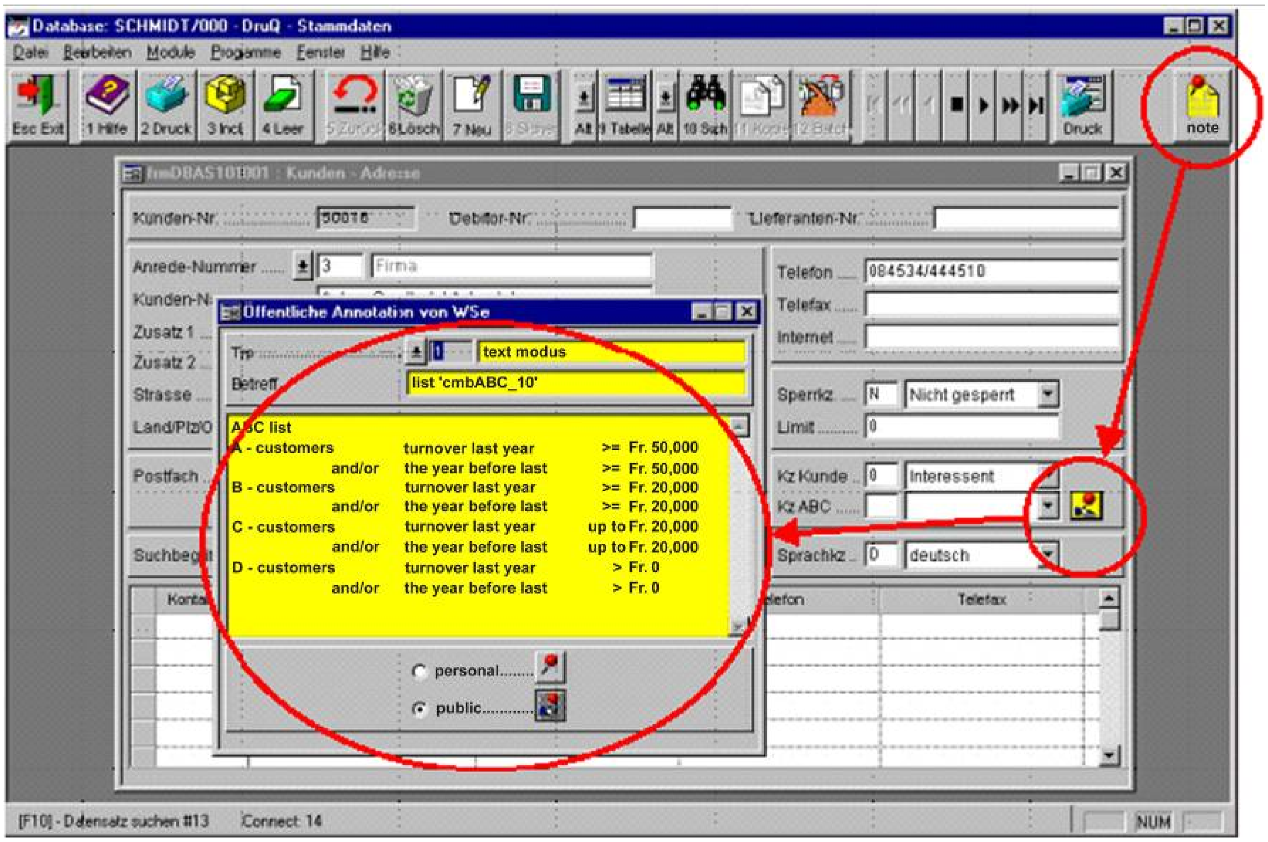

Figure 1: Annotation button (“note”) in the tool bar, annotation icon and annotation edit window

The annotation extension has been integrated in a printing shop software application and has been installed for 2 years at about 80 work places in 3 companies. For evaluating the usage of the annotations an evaluation function was implemented to store the creation and reading of the annotations. Interesting for the evaluation of the creation of annotations was the author, the date compared to the introduction of the software into the department, the length of the annotation and the private or public type of the annotation. We have expected a high proportion of annotations at the beginning of the software lifecycle, first public annotations by administrators and then also public and private once by users. Interesting for the usage of annotations was the frequency of the usage, the time of the usage and the type of annotations. We have expected that most of the annotations read were public annotations and that most of the annotations were read at the beginning of the lifecycle of the software some later than the peak of the creation of the annotations.

The following table shows the quantitative distribution of annotations in the three companies with different time periods of software lifetime.

company 1 (2 years)		company 2 (1 year)		company 3 (1 week)		total	
employees	annotations	employees	annotations	employees	annotations	employees	annotations
20	56	23	48	35	24	78	128

Table 1: 128 annotations created by 78 involved employees in 3 companies

The table shows that in all companies a substantial number of annotations has been created. Even in the case of the third company in only one week time 24 annotations has been set. Compared to the masks of the application on average 1 annotation was set for permanently used masks and 1 annotation was produced for 3 of all possible masks of the application. There are differences in the number of created annotations between the companies. The differences can in part be explained by the time duration of the observation period: in case of long usage periods more annotations has been created. Another factor could be the training of the users: not all users were trained in all companies to use the annotations; in companies 2 and 3 only the administrators has been trained as multipliers. No annotation was created by users who were not explicitly trained about the concept of annotations.

company 1		company 2		company 3		total	
public	private	public	private	public	private	public	private
41	15	22	26	12	12	75	53

Table 2: Distribution of the created annotations to the public vs. private types

Table 2 shows that compare to private annotations more public ones have been created open for the complete user community in the respective company. Annotations are thus not only personal memos but are also used for communicating information to other users. Information include tips for handling the application, instructions to perform the tasks and conventions for classifying products and services for classes of clients of the printing shops.
Most of the annotations (68%) were created by the administrators of the three companies. But also users have created annotations (32%). Most of the annotations of the administrators were public; most of the non-administrative users have created private annotations. Most of the users have only created public or private annotations – saying that possibly the differential meaning of the type concept has not been understood or not been seen relevant.

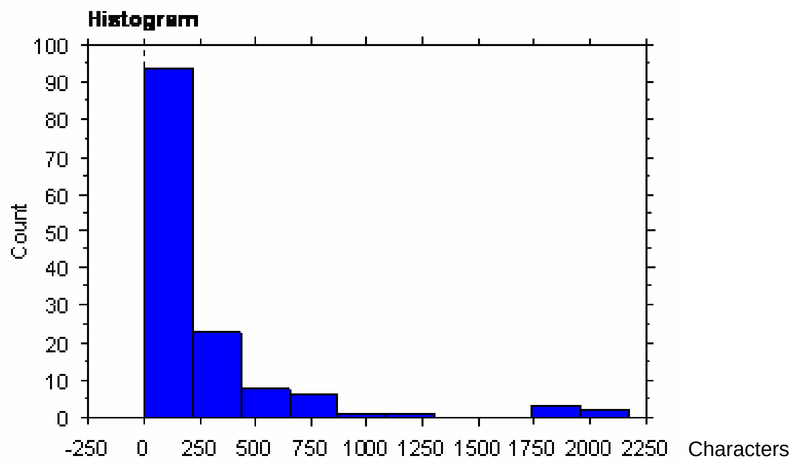

Figure 2: Number of characters included in the annotations

Most of the annotations contained only up to 250 characters (about 2 ½ lines in this paper). Annotations in practice can be perceived as short messages with comprehensive information and not as complete dossiers. There were only a few annotations that contain longer explanations (up to half a page of this paper).
What holds for the length of the created annotation also holds for the time spent to read the annotations. The following figure shows the results of one of the companies.

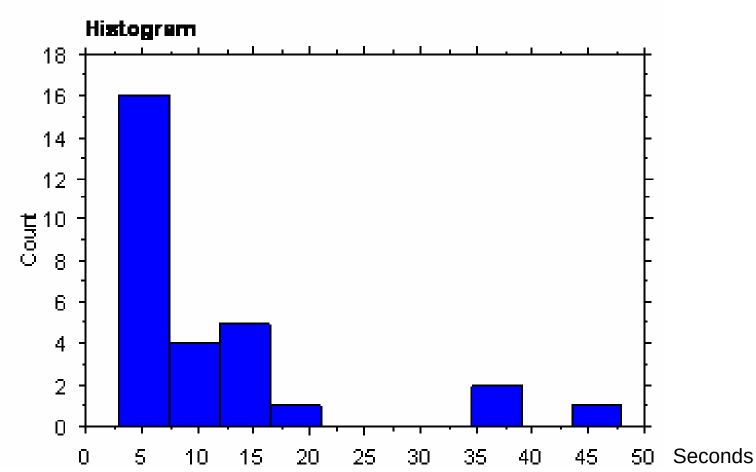

Figure 3: Number of seconds spent to read the annotations

The modus of the reading time is 5 seconds with only some cases up to 15 seconds and a very few once with longer time. This time is similar to the time needed to read this figure: 5 seconds to get the message ("majority with 5 seconds reading time") and 20 seconds to study the details with the specific scaling parameters and about 40 seconds to answer a phone call or to take a sip of coffee in between.

The time of the creation of annotations was supposed to concentrate on the beginning of the lifecycle of the software application. The following figure shows the frequency of creation dates in two companies with the longest usage times (company 1 and 2).

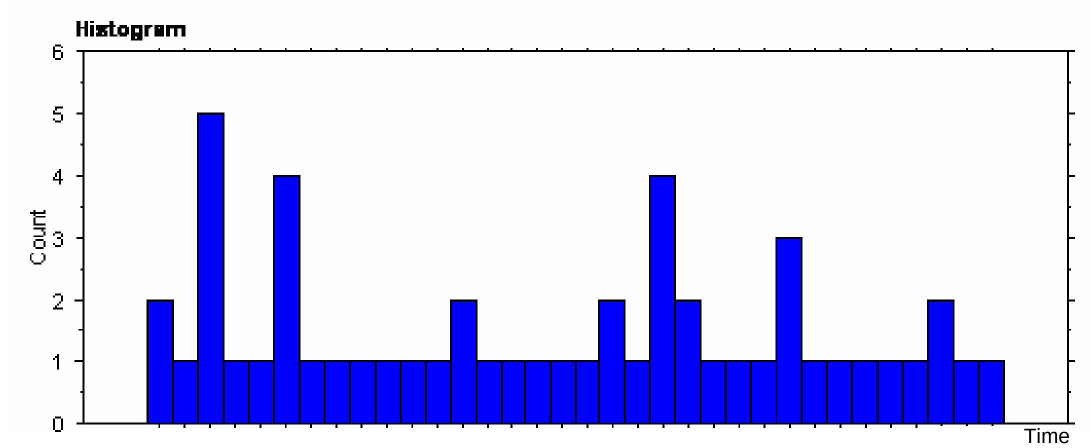

Figure 4: Frequency of the creation of annotations within the time period of two years

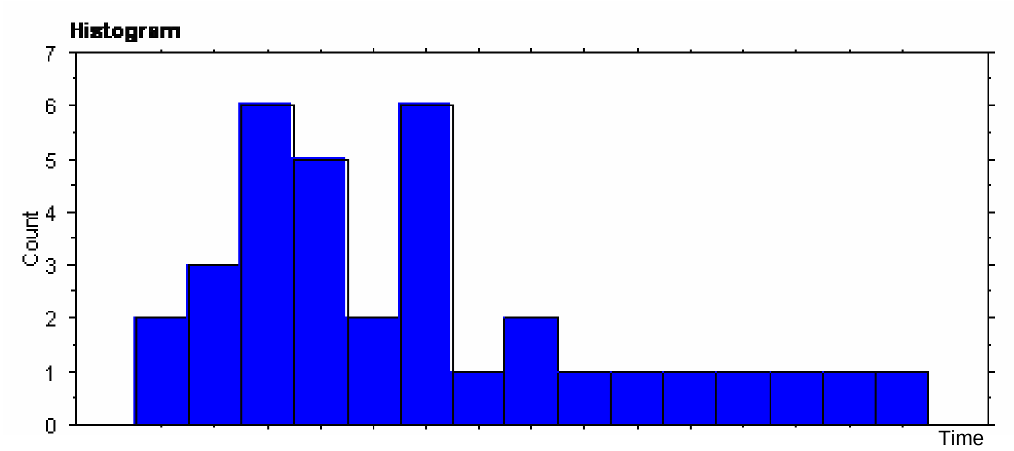

Figure 5: Frequency of the creation of annotations within the time period of 1 ½ years

The results show that during a longer time (at least beyond the explicit rollout time) annotations were created. Administrators concentrated their annotations to the early implementation time of the system, possibly because it is their job to support the understanding and handling of the system. Annotations of administrators later in the usage period summarized problems and gave solutions in cases where the administrators realized typical questions of the employees (cf. FAQs in the WEB). Employees entered annotations also later in the time line; at the beginning of the usage time they had enough to do to grasp the fundamentals of the new system; they tried to explore and exploit the full potential of the system later in the time line and supported this secondary circle of knowledge acquisition by private or public annotations.

Reflecting annotations for learning support in the work context is one direction to embed learning in the process of work. There is also another direction to embed annotations into an activity process. Krause and Keindl have shown in a study about the design of learning culture and development of competency based multi media that means are important to store and exchange experiences and knowledge abut specific cases to enrich the learning process (Krause and Keindl 2005). They report that more than ¾ of the sample of 41 cases said that documentation and communication of specific real world cases in the learning platform is important or very important.

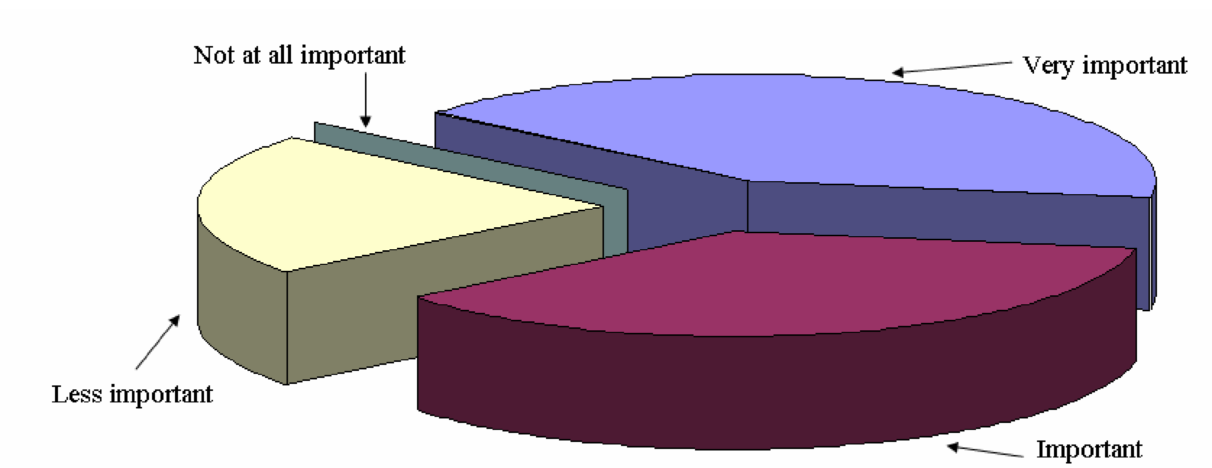

Figure 6: Importance of documentation and communication of specific real world cases in the learning

The results of the findings have shown that integrating reflection means for the learning process are accepted and welcomed by the users in both directions: situated learning during the task accomplishment and authentic work cases of the community during the learning process.

3.2 Contextualised episode recordings

Beyond the simple annotations for notes at relevant elements of the interface of the application contextualised episode recordings help the user to memorize the sequence of actions during the authentic work. While or after facing a problem or while or after realizing a solution the user can use the episode recorder in order to store the critical episode for later use. Later use can mean own later inspection of a critical episode for recognition or the communication about the critical episode with somebody near the workplace or at remote help desks. Episode recordings are similar to remote shared screen or screen recorders like Lotus ScreenCam. The difference is that the episode recorder stores the individual interaction sequence as a continuous temporary window of, say, 30 minutes or two hours. During the work of the user in case of critical episodes s/he can address the history of the dialog sequence whenever needed to inspect a critical incidence. This critical incidence the user can capture and store in what we call an episode studio as containment for relevant episodes for this user. With classical tools the user has to know in advance, when to start the recording, i.e. when a relevant episode will occur, or s/he has to repeat the input/output sequence of the critical episode once s/he has realised that an error occurred or a relevant solution has been found.
The contextualised episode recorder stores the interaction sequence of the critical incidence together with context parameters to help the user to recover the episode in similar situations later. Of course the user can enter a name for the episode but automatic context parameters help the user to recover the episode among a growing number in later situations. Context parameters stored automatically together with the user name and the explicit entered episode name are: the application, the document, the function and the date of recording the episode. These context parameters help the user to retrieve the episode in later similar situations.

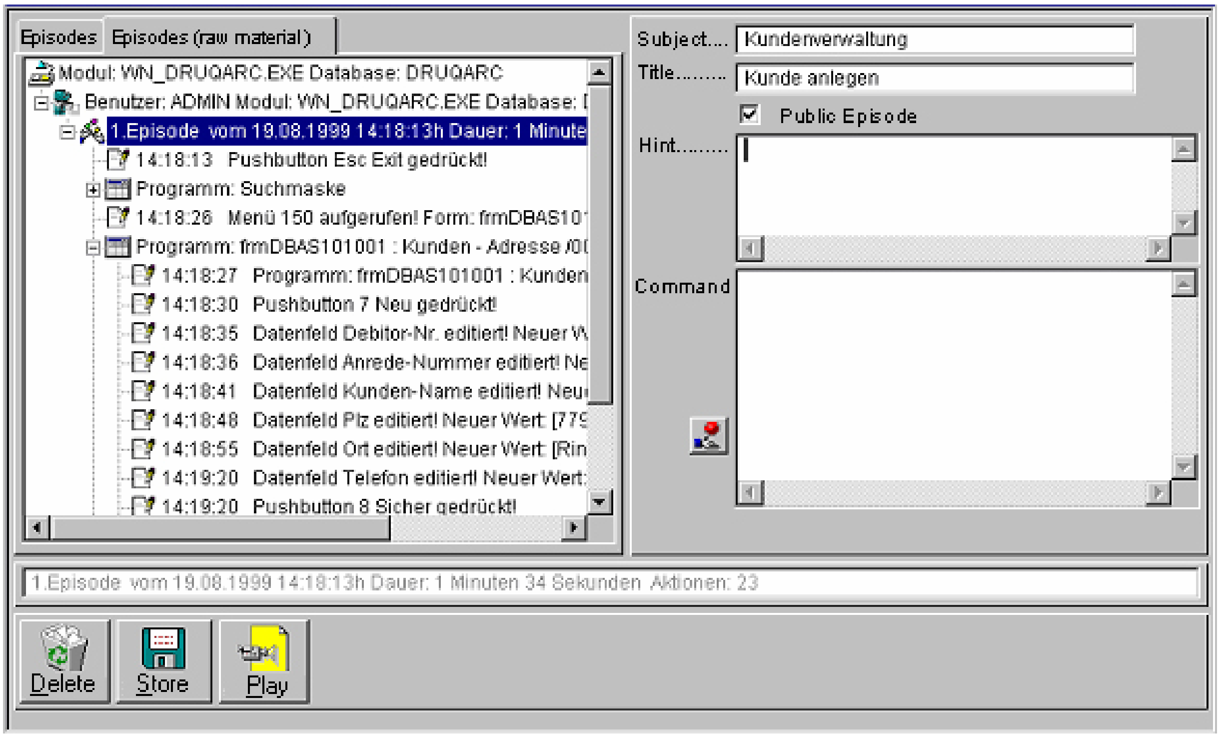

Figure 7: Editing window of the episode recorder

The editing window shows at the left side a sequence of interaction steps of the user "ADMIN" in the time period between 14:18:13 and 14:19:20. At the right side a user defined name and references can be entered and the episode can be published for the community of users. At the bottom of the editing window the user can delete, store or play the episode currently selected in the episode window.
After creating a set of contextualised episode recordings the user can directly "play" the episode from the episode studio if the user knows which episode is of relevance for the current situation. If the user does not remember, which

episode can help to solve a current problem s/he can use the context parameters of the episode like the application, the document, the function or the date when first experienced the problem.

In our research group we have extended the idea of contextualized episode recordings for two other applications to support not only classical desktop based work but also

- Mobile work of maintenance personnel far from desktop workplaces and
- Distributed learning during field trips of students collaborating with their peers in the classroom, a lab or a library.

For mobile maintenance personnel the heterogeneity of installations and machineries gets more and more complex. The need for explanations of tasks and procedures grows. Instructions based not only on manuals but also on mobile computers like PDA or tablet-PC support the person conducting the maintenance task. For situations where procedures are not covered by standard instructions the maintenance personnel develops individual solutions on his own or in (tele-)cooperation with other personnel (peer-experts). A prototype has been developed to show the idea for a mobile instructor: an iManual for the navigation system of a high-tech car (Klann, Humberg et al. 2005). The prototype integrates mobile instruction and help components for the customer and the service personnel with the product development and refinement process based on the problem and solving records of the users. Such solutions can also be documented by a mobile recorder that we call Mobile Collector (Kravcik, Kaibel et al. 2004). The user can record his work on the machine primarily used for the instruction and comment the critical incidents and solutions in the modality s/he prefers. Thus, the mobile collector can be understood as an extended instruction computer (PDA or tablet-PC) that includes multimodal input techniques like pen based handwriting, video camera and microphone. In later situations the maintenance personnel can refer to the recording by him-/herself or can also give access to other personnel of the company.

Learning more and more takes place at different places. For instance, students take part in excursions (field trips). Field trips are expensive in terms of time and money so that they are some times restricted to a limited number of students. To integrate the rest of the students in the classroom or to integrate specialists in a lab or in a library as different roles in a scenario of distributed exploration and learning the Mobile Collector can be used to record the activities and the exploration results of the field trip students and transmit the data to the remote peers who can study or also analyze the data transmitted by the field trip students and send back instructions for further research activities in the field. The configuration supports the cooperation and real time exploration between several individuals or several groups of students.

The applications of the Mobile Collector have not yet been evaluated in detail. There were about 15 field trials conducted with distributed teams of students with different configurations of exploration, documentation and communication in the field and in the classroom concurrently (Kravcik, Kaibel et al. 2004).

4 Conclusion

The presented concept of annotation and recording facilities for supporting the integration of learning and task accomplishment has been illustrated in different directions. One direction shows the benefit of annotation and recording facilities for the learning process where static notes can be administered for elements of the application (functions, icons, entry fields etc). Elements and features of these annotations have been illustrated and the empirical evidence showed clear acceptance of the users in authentic field trials. The episode recording implementation has been developed because not all of the long-term memorisation of procedures can be captured in text, figures or audio comments. Processes can best be represented by screen recordings of critical episodes in the work procedure. For the reuse of such critical episode recordings different retrieval methods have been included. The potential of the presented concepts could not be proved by sufficient empirical experiments. But the results of several explorative field studies showed the value of the methods if they are integrated into the authentic work and learning context. More research is needed to find out the relative strength of the help users and user communities receive from the presented facilities compared to the effort needed to develop the tools in the production context and the effort to apply the tools in the work context.

Acknowledgment

I have to thank Wolfgang Bihl who developed the annotation and recording tools and collected the data of the field trials. I have also to thank Milos Kravcik and Andreas Kaibel for data about Mobile Collector and Markus Klann for data about the iManual.

References

Ackerman, M. S. & T. W. Malone (1990). "Answer Garden: A tool for growing organizational memory." Proceedings of the Conference on Office Information Systems 11: 31-39.

Bannon, L. J. (1986). Helping Users Help Each Other. User Centered System Design. New Perspectives on Human-Computer Interaction. D. A. Norman and S. W. Draper. Hillsdale, N.J., London, Lawrence Erlbaum Associates: 399 - 410.

Brezizinski, R. (1987). When it's Time to Tear Down the Info Center. Datamation 33 (1988), 21, 73 - 82.

Brockmann, R. J. (1990). Writing better Computer User Documentation: From Paper to Hypertext. New York: John Wiley & Sons.

Carroll, J. M. (1990). The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill. Cambridge, MA: MIT Press.

Dugas, M., M. Batschkus, et al. (1997). CBT in der Medizin mit Intranet-Technologie - Erfahrungen im Klinikum Großhadern. In: Conradi, H./R.Kreutz/K.Spitzer (Hrsg.): CBT in der Medizin. Methoden, Techniken, Anwendungen. Proceedings zum Workshop in Aachen 6. - 7. Juni 1997, 17 - 20.

Fox, T., G. Grunst, et al. (1994). HyPLAN: A Context-Sensitive Hypermedia Help System. Adaptive User Support. Ergonomic Design of Manually and Automatically Adaptable Software. R. Oppermann. Hillsdale, New Jersey, Lawrence Erlbaum Associates: 126 - 193.

Horton, W. K. (1990). Designing & Writing Online Documentation: Help Files to Hypertext. New York: John Wiley & Sons.

Klann, M., D. Humberg, et al. (2005). "iManual - Mobile Endgeräte als kontextsensitive integrierte Bedien- und Hilfesysteme." Wirtschaftsinformatik 47 (1): 36–44.

Krause, A. & K. Keindl (2005). Wissenschaftliche Begleitung von Praxisprojekten zur Lernkulturgestaltung und Kompetenzentwicklung durch multimediale und netzgestützte Techniken. Berlin, Core Business Development Institut für Produkt und Prozessinnovation: Forschung – Beratung – Training.

Kravcik, M., A. Kaibel, et al. (2004). "Mobile Collector for Field Trips." Educational Technology & Society 7 (2): 25-33.

Liechti, M. (1988). Das "Information Center" - wichtige Supportstelle im Betrieb, 1. Teil, io Management-Zeitschrift 57 (1988), 574-575.

Moning, U. & B. Winkelmann (1993). Entwicklungsphasen von Information Center, Ergebnisse einer empirischen Untersuchung über betriebliche Endbenutzerunterstützung in der Schweiz. In: Wirtschaftsinformatik, 6 (1993), 532-541.

Nardi, B. A. (1993). A Small Matter of Programming. Perspectives on End User Computing. Cambridge MS, The MIT Press.

Norman, D. A. & J. C. Spohrer (1996). Learner Centererd Education. Communications of the ACM. 39: 24-27.

O´Malley, C. E. (1986). Helping Users Helping Themselves. User Centered System Design. New Perspectives on Human-Computer Interaction. D. A. Norman and S. W. Draper. Hillsdale, N.J., London, Lawrence Erlbaum Associates, Publishers: 377 - 398.

Paul, H. (1995). Exploratives Agieren. Ein Beitrag zur ergonomischen Gestaltung interaktiver Systeme. Frankfurt am Main, Peter Lang Verlag.

Santhanam, R. & S. Wiedenbeck (1993). Neither novice nor expert: the discretionary user of software. International Journal of Man-Machine Studies 38 (1993), 2, 201 - 229.

Schank, R. C. & A. Kass (1996). A Goal-Based Scenario for High School Students. Communications of the ACM 39 (1996), 4, 28 - 29.

Williams, T. R. & D. K. Farkas (1992). Minimalism Reconsidered: Should we Design Documentation for Exploratory Learning? SIGCHI Bulletin 24 (1992), 2, 41 - 50.