Journal of Universal Computer Science, vol. 11, no. 3 (2005), 394-409
submitted: 31/1/05, accepted: 7/3/05, appeared: 28/3/05 © J.UCS

Using Weblogs for Knowledge Sharing and Learning in Information Spaces

Eric Ras
(Fraunhofer Institute for Experimental Software Engineering, Germany
eric.ras@iese.fraunhofer.de)

Gabriela Avram
(Centre de Recherche Public Henri Tudor, Luxembourg
gabriela.avram@tudor.lu)

Patrick Waterson
(Fraunhofer Institute for Experimental Software Engineering, Germany
patrick.waterson@iese.fraunhofer.de)

Stephan Weibelzahl
(National College of Ireland, Ireland
sweibelzahl@ncirl.ie)

Abstract: There are various Knowledge Management Systems available currently and designed to support knowledge sharing and learning. An example of these are "Experience-based Information Systems" in the domain of Software Engineering, i.e., Information Systems designed to support experience management. Lately, these have become more and more sophisticated from a technical point of view. However, there are several shortcomings that appear to limit the input, the content of these systems and their usage. The problems identified in this paper relate to knowledge acquisition, learning issues, as well as to the users' motivation and trust. We introduce an approach meant to enhance the content of the experience base and improve learning from experiences within information spaces, namely weblogs that are maintained during daily work and serve as input for both an experience base and for an information element base. In order to enhance learning, a pedagogical information agent is envisaged for retrieving suitable experiences to be further enriched with additional information elements and produce micro-didactical learning arrangements. In addition we consider the relevance of motivation and trust issues. An empirical study demonstrates that using weblogs for such an approach is feasible.

Keywords: Experience-based Information System, wiki, weblog, pedagogical information agent, information space, micro-didactical learning arrangement
Categories: A.1, D.2, H.4, J.4, K.3

1 Introduction

Within the Information and Communication Technologies sector, continuous competence development is essential in order to combat the increasing flood of information, the rapid deterioration and ageing of knowledge and to face the continuously changing requirements for problem understanding and solving.

In the domain of Software Engineering, software development can be considered as a human-based knowledge intensive activity. Together with sound methodology and technology, the success of a software project depends strongly on the knowledge and experience brought to the project by its developers.

In the past, developers have mostly depended upon tacit knowledge. This resulted in problems when experts left a project and new developers entered. The tacit knowledge was not kept within the organization, and therefore the learning curve for novice developers resulted in a significant lowering of the software quality. Hence, Knowledge Management Systems and Experience-based Information Systems have been developed to address the problem of knowledge loss and to improve knowledge sharing in general. Knowledge sharing can be seen as a type of informal learning where knowledge is imparted and obtained unconsciously. Informal learning relates to the lifelong process in which every person acquires and accumulates knowledge, skills, attitudes and insights from daily experiences and exposure to the environment [Marsick and Watkins 02].

In this paper we review the shortcomings of EbIS and propose an approach for solving some of these problems. The approach focuses on how to motivate people for informal knowledge sharing within weblogs and on content elicitation from these weblogs in the knowledge acquisition phase for learning purposes. Our empirical studies demonstrate that weblogs can actually be used as an alternative source for enriching both the experience base of an EbIS and for producing learning content.

The two following sections provide a short introduction to Experience-based Information Systems, Wikis, and Weblogs.

1.1 Experience-Based Information Systems

The field of Experience Management, as a sub-field of Knowledge Management, aims to support the management and transfer of relevant experiences. The software system used for managing, storing, retrieving, and disseminating these experiences is known under the name of Experience-based Information System (EbIS). EbISs are a special type of Knowledge-based Systems (KBS) that build their intelligent features on the explicit representation of knowledge or experience [Tautz (00)].

Knowledge and experience. While "*knowledge* is the range of learned information or understanding of a human or intelligent information system, *experience* is considered to be knowledge or practical wisdom gained through human senses, from directly observing, encountering, or undergoing things during the participation in events or in a particular activity" [Tautz (00)]. We actually share the position of Stenmark [Stenmark 01] and consider the usage of the term "knowledge" for information stored in a computer inappropriate. Tacit knowledge can in fact exist only in the heads of people, and explicit knowledge is actually information. However, the terminology used in the theory and practice of KBS considers knowledge to be information stored together with its context, and we follow this convention throughout this paper.

There were some interesting developments striving to support lifelong learning by eLearning integration within experience-based systems in organizations [Althoff and Pfahl (03)] Nevertheless, they are seldom adjusted to the learning demands of the individuals, which are very diverse, and they encounter difficulties in quantifying factors. On one side, there are learning demands that are growing and becoming more

complex every day, and on the other, the learning sources, similarly numerous, but not always fit for covering the expressed needs.

Knowledge sharing, as an integral part of Knowledge Management, was lately one of the most frequently discussed topics amongst managers and technologists. Knowledge sharing is on everyone's lips but unfortunately it is seldom properly translated into practice. The best way to ensure knowledge sharing is often understood to be the acquisition and storage of knowledge in knowledge bases, followed by countless and costless sharing. After capturing knowledge in the knowledge base, it should have been possible for everyone else to "come and drink from the source"- at anytime and from anywhere.

One of the domains where KBSs and EbIS in particular as a source of learning have been profitably implemented is Software Engineering, a rapidly changing, knowledge-intensive business involving many people working in different phases and activities within the product lifecycle. However, organizations often encounter problems identifying the content, location, and use of knowledge [Rus and Lindvall 02]. Other problems are related to the acquisition of new knowledge, to the fact that already stored knowledge is not suitable for learning purposes, and that maintenance and evaluation of the already documented knowledge are not done appropriately.

1.2 Social Software

In this paper, we propose the deployment of content management systems intended for both individual and workgroup use, such as wikis and weblogs, for capturing basic information, knowledge, and experiences [Angeles 03]. We will show how the input from wikis and weblogs can be used as a source for enriching the experience base (EB) of an EbIS and for creating information spaces for learning purposes.

As an introduction, we present some of the concepts used in the paper. Wikis and weblogs are software applications belonging to the social software category - a particular sub-class of software concerned with the augmentation of human social and/or collaborative abilities through structured mediation [Coates 03].

A *weblog*, or simply a blog, is a web application which contains periodic posts on a common webpage [Wikipedia 05]. These posts are often, but not necessarily, in reverse chronological order. Such a website would typically be accessible to any internet user. Some are maintained by single authors, while others have multiple authors. Many weblogs enable visitors to leave public comments. The format of weblogs varies, from simple bullet lists of hyperlinks, to article summaries with user-provided comments and ratings. Individual weblog entries are almost always date and time-stamped, with the newest post at the top of the page. Because links are so important to weblogs, most blogs have a way of archiving older entries and generating a static address for individual entries; this static link is referred to as a permalink. The latest headlines, with hyperlinks and summaries, are offered in weblogs in the RSS or Atom XML-format, to be read with a feed reader.

A *wiki* is a website (or other hypertext documents collection) allowing users to add content, but also allows anyone to edit the content. *Wiki* also refers to the collaborative software used to create such a website [Wikipedia 05].

The rest of this paper is organized in the following way: Firstly, current shortcomings of Experience-based Information Systems are reviewed (Section 2). Secondly, we propose an original approach addressing some of these shortcomings

(Section 3). Finally, in Section 4, we present the findings of two empirical studies investigating informal knowledge sharing through weblogs and wikis as an alternative to provide content for information elements, and hence to reduce the effort of producing learning content.

2 General Shortcomings and Issues to Be Addressed

This section describes three interdependent issues related to the update and use of experiences included in an Experience Base (EB), namely the elicitation of new experiences, learning based on these experiences, and issues related to motivation and trust in using an Experience based Information System (EbIS).

One of the most difficult problems in maintaining an EB is the elicitation and documentation of new experiences. In order to obtain the best results, knowledge should be captured immediately after the new experience's occurrence. To be included in the EB, the new experience must be described in a structured way, and information on the context of the specific experience must be also added. This is an activity that requires a lot of effort and dedicated skills [Hellström, Malmquist, and Mikaelsson 00]. The utility of an EB grows with the amount and quality of experiences included in it. An almost empty EB is useless, since the probability of retrieving meaningful experiences when needed is very low.

Considering the usage of KBSs in general, and of EbISs in particular, learning is considered to be a fundamental part of KM, since employees have to internalize (i.e., learn) shared knowledge before they can use it to perform specific tasks [Rus and Lindvall 02]. KBSs make the assumption that the problem of continuous competence development can be partially solved by using intelligent retrieval mechanisms and benefiting from innovative presentations of retrieval results. As a result KBSs focus mainly on knowledge acquisition, storage, and retrieval and less on the learning processes themselves, the integration with the work process, and on the personal needs of users.

Last but not least, to motivate and to convince users to contribute and to make use of the EB is not an easy task. Sharing knowledge can be perceived as being short of advantages for oneself and even dangerous. Additionally, contributing to the EB is time and effort consuming. There is a certain resistance that has to be coped with.

A detailed analysis of the shortcomings of current EbISs based on the literature [Hellström, Malmquist, and Mikaelsson 00], as well as on our own experience, led to the description of several problems related to either the quality of the EB content, to learning issues, or to psychological factors such as motivation and trust.

2.1 Issues Related to the Content of the Experience Base

The quality of the EB content is obviously a crucial point for various reasons. Firstly, the elicitation of knowledge from experts and users causes potential problems, since new experiences must be first identified in order to achieve adequate coverage. If there are no triggers to draw the attention upon recent experiences, important insights may be lost. There is no other alternative than having individuals and teams deciding to report what might be useful for others and finding the appropriate level of granularity – such a task involves value judgment. Of course, precise rules regarding

when and how this should be done can be imposed, but the decision belongs to real people. Human judgment is subjective, and some could decide that minor experiences are important and must be shared, while other could overlook important experiences considering them insignificant or unreliable.

Secondly, in most systems the elicitation procedure imposes some strict rules for including a new experience usually resulting in time-consuming and complex tasks to be performed by people disposed to describe it. Not only that highly structured reporting is time consuming, but there is also a trade-off of structured report vs. flexibility: the more formalized and standardized the experiences have to be reported, the better the retrieval, but the lower the ability to suite the different preferences of those who are supposed to re-use these experiences.

Thirdly, the quality of the reported experience highly depends on the individual communication skills of the contributor, e.g., the ability to structure the content, to formulate the experience with accuracy, and to describe it properly according to the needs of the target audience. The whole process is one of mediated communication: (i.e., people are leaving messages to other people). The EbIS only serves as an intermediary. Further, the lack of ability in coping with the ambiguity of experiences (solutions that worked in one case and failed in others, for example) could prevent both the contribution to the EB or their use afterwards.

Fourthly, the maintenance of the EB requires regular evaluation of all experience packages and removal of outdated entries. Storing several contradictory solutions for a sole problem, originating from different persons at different points in time is a source of confusion and mistrust. The capacity to retrieve previous related experiences exists in EbISs, but it requires time and effort to review them. The risk of applying an outdated experience is high, if experiences do not have an attached expiry date, and in many occasions these repositories become a sort of graveyards – some experiences are added, but nothing is ever thrown away. With the continually growing size of EBs, it is difficult to keep an overview in order to connect related packages and to avoid inconsistencies.

2.2 Learning Related Issues

Even if the EB has a high coverage and precision, the experience packages might still be inappropriate for learning due to two reasons: the inappropriate quality of the content for learning purposes and the fact that learning processes are not explicitly addressed by the EbIS.

In most of the cases, Software Engineering experiences are documented by experts with a deep knowledge of the domain. Learning from knowledge shared by an expert raises several issues (see [Ras, 04] for a summary). Often learners need additional information about the subject domain, since experts provide an experience description without adding extensive explanations of the experience itself, the conditions and prerequisites that let the experience happen, and the experience's context. Knowledge acquisition always depends on the existing structure of human memory. The qualitative difference, i.e., the organization of knowledge at the experience provider and at the consumer makes the transfer of knowledge between different levels of expertise extremely difficult [Ericsson, Krampe, and Tesch-Römer 93]. Another problem is that expert knowledge is somehow "routine". This makes it

difficult for experts to document experiences appropriately and to make them reusable for others.

The utilization of an EbIS is usually problem-driven, i.e., a problem arisen during the completion of a Software Engineering task motivates the software developer to check for suitable experience packages and solutions in the EB. When reusing an experience package, a developer is usually engaged in active problem-solving while reading, understanding, abstracting or instantiating the experience package, and trying to apply the gathered knowledge to the real problem situation. Ideally, software engineers could learn effectively from experiences when all four phases of Kolb's *Experiential Learning Circle* [Kolb (84)] are passed: making a concrete experience, observing and reflecting about the occurrence, forming abstract concepts, and testing these concepts in new situations. When a software engineer is documenting an experience for later reuse (i.e., this is done usually by creating abstractions), he or she profits from being involved in the situation that leads to the experience, and their own observation and reflection about the happening. When a software engineer other than the experience's provider wants to reuse this documented experience, he or she will lack of specific knowledge about the event that led to the experience, and the knowledge that results from observation and reflection. Hence, EbIS should focus on the delivery of appropriate content in addition to the experience package in order to support knowledge construction as described in Kolb's learning cycle.

2.3 Motivation and Trust Related Issues

In order to use and contribute to the EB, users need to be motivated. A variety of problems are associated with the use of EbIS, these including the fact that sharing knowledge is perceived as dangerous if competitors could use the shared knowledge. Potential contributors may have concerns that they are supplying others with knowledge without any profit. Moreover, both contributing to and consulting the EB can be seen as time and effort consuming activities. However, incentives seem to help less than the statutory obligation to provide feedback both during and at the end of each project [Stenmark 02].

These problems are further compounded by the fact that users rarely want to solely share their knowledge and frequently are motivated to start by re-using somebody else's experience. In other words, it is important to facilitate the initial usage of the system by providing a sufficient fundament of relevant experience packages.

Finally, users might also refuse to apply an experience package, lacking confidence in its validity. It might be unclear whether a packaged experience is still up-to-date or whether applying it might involve a certain risk. It is also possible that the relationships to other experiences were not appropriately captured. The quality of packaged experiences could also be influenced by contradictory interests: employees are much more inclined to share positive experiences, because bringing to attention their failures could have a negative connotation. The "not invented here" syndrome can be another cause of reluctance in reusing experiences, and to overcome it, there must be a shift in the culture of the organization [Rus and Lindvall 02]. Particularly for global organizations, work-related cultural differences related to power, individualism, or gender could also have a negative impact.

3 Possible Approaches

In the previous sections we presented content, learning, motivation, and trust related problems of EbISs. In this section, we are proposing an original approach for addressing some of these issues. Figure 1 presents the framework of our approach.

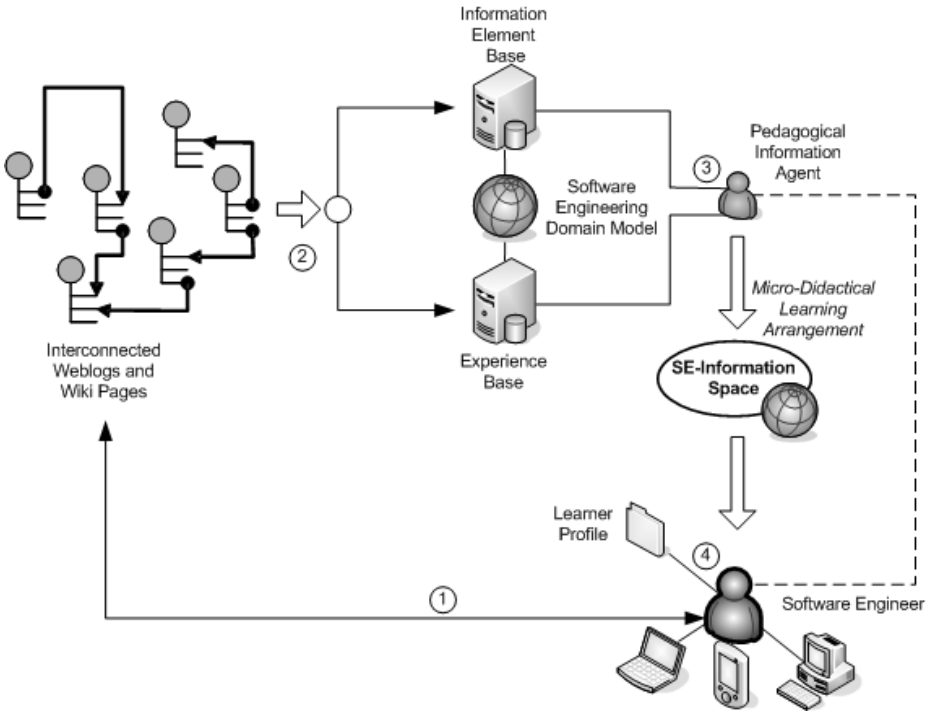

Figure 1: Framework of the approach

The attempt to capture knowledge (considered as information-in-context) from weblogs provides contributors with the freedom and flexibility to add whatever experiences they consider to be relevant. This also brings about the opportunity to capture new experiences immediately after their occurrence (1). The content of the experiences selected for inclusion in the EB has to be later detailed and structured by a specialized team. If considered relevant, new content will be either stored in the experience base or used for enriching the information element base (2).

A pedagogical agent is taking responsibility for creating and providing the user with an information space by creating micro-didactical learning arrangements (i.e., enriched experiences with information elements) (3). Stored learner profiles ensure the proper structuring and presentation of information elements in the information space. An atmosphere of trust will be supported as a result of the mutual exchange of experiences (4). A more detailed description of this framework is provided in the following sections.

We claim that the use of weblogs for recording knowledge is a feasible alternative for supplying new experiences to be added to the EB and to be used in providing content for new information elements to be added to the IEB**.**

3.1 Experience Elicitation Related Approaches

The EM approach is typical for highly centralized KBSs. At the other end, the alternative solution is offered by distributed knowledge management tools such as weblogs and wikis, which focus more on interaction amongst people.

A possible scenario for addressing some of the problems related to experience elicitation is:

- complementing the EB, which is a centralized experience repository, with less structured distributed repositories – weblogs – created and maintained by individuals or work groups;
- through regular observation, the meaningful experiences posted in weblogs could be identified; they could be included in the EB after being documented and restructured by a specialized team;
- some other weblog posts or wiki articles could be identified as a source of information elements to be extracted and stored for learning purposes.

Of course, the usage of very simple and relatively affordable software tools for sharing knowledge available to any employee at any moment in time could overcome some of both the technical and the psychological barriers to sharing knowledge. However, in practice their availability is not sufficient for changing the employees' habits or encouraging them to routinely share knowledge. In order to support such an attitude change, we defined a special role: that of *knowledge brokers* – a similar role was mentioned in [Hellström, Malmquist, and Mikaelsson 00]. Their task is to encourage and support the use of such distributed knowledge repositories. They should continuously assist and support the employees to adapt to the new situation, acting as facilitators, moderators and coaches and providing examples, success stories and access to technical support. The identification of experiences suitable for inclusion in the experience base, their restructuring and documentation will be performed as before by a specialized team, supported by the knowledge brokers. The identification of information elements fit for learning purposes and the annotation according to a Software Engineering domain model has to be taken over by a specialist with a solid background in both the domain of expertise and in learning technologies.

3.2 Learning Related Approaches

The approach that we currently develop focuses on the combination of plain experiences and information elements (IE). IEs could be term definitions, application examples, explanations, Software Engineering principles, context information, or information about certain persons etc. As described in the previous section, content for IEs can be identified within weblog posts and wiki articles. The approach aims at improving the suitability of the delivered input for learning by embedding the experiences in so-called micro-didactical learning arrangements, i.e., information

spaces with an experience as central element. While usually the experience packages in EbISs are just retrieved and presented without any further explanation, we propose to augment the presentation with additional information elements to facilitate a better understanding and applicability of the experience packages.

An information space is created by a pedagogical information agent. Pedagogical agents are a special type of information agents: they put their emphasis especially on the mediation of information by taking into account the learning preferences stored in the learner's profile, such as preferred learning styles, presentation modes etc. Information agents possess skills such as retrieving, analyzing, manipulating, and fusing heterogeneous information as well as visualizing and guiding the user through the available individual information space [Klusch 01].

The pedagogical information agent is set off either top-down, by the software engineer through explicit queries in the EbIS, or bottom-up through monitoring information sources for the occurrence of particular events (e.g., mining for specific keywords entered by the software engineer while typing a document or writing a program).

Currently, a pedagogical information agent prototype for supporting the refactoring process in Software Engineering is being developed [Rech, Ras, and Jedlitschka 04]. Refactoring is a process for removing or reducing software quality defects and thus improving the quality of software systems. When the agent finds a defect in the code, a query is forwarded to the EbIS to retrieve similar cases and previous experiences related to this type of defect. The agent creates a micro-didactical arrangement according to the learning profile of the software engineer. This arrangement is a subset of the information space. The information space could offer further information elements that go beyond the scope of the retrieved experience to the user. For example, information elements explaining the background of defect reduction or software quality in general. Such an information space consists of several pages with links forming a hypermedia network. The pages are composed of information elements retrieved from the IEB. Agent technology allows us to adapt the information space dynamically during run-time, for example, while the user is browsing through the space and removing the defect. While the learner is accessing information elements in the information space, the agent is observing his/her activities and adapts the learning space and the learning profile accordingly.

3.3 Motivation and Trust Related Approaches

Encouraging people to share their knowledge has long been a subject for discussion amongst academics and practitioners involved with the subject of Knowledge Management. Similarly, one of the main barriers to knowledge sharing relates to the degree to which individuals trust the information they can retrieve from a knowledge base, as well as the extent to which they feel their own contributions will not be misused. In a review of some of the factors leading to successful knowledge sharing, [Hall 01] makes a distinction between "enabling" conditions that improve the environment for knowledge sharing and the provision of "explicit" and "soft rewards" within these environments.

One example of an enabling condition mentioned by Hall is making knowledge sharing a key responsibility of staff. There are many ways in which this can be achieved including the provision of proactive training and project debriefings

associated with use of an EbIS, as well as other strategies such as mentoring and assisting users. Hall cites a case study conducted at Citibank that demonstrated that assigning specific responsibilities to individuals was more effective in bringing about knowledge sharing, as compared to expecting people to make contributions as part of a larger team effort alone [O'Dell 98].

Providing rewards for knowledge sharing has also proved in many cases to be an effective mechanism that can be used to bring about effective and widespread knowledge sharing. These can take the form of "explicit" rewards such as including in criteria for promotion and appraisal systems the degree to which employees have submitted and shared their past experiences within their company. In the case of "soft" rewards it may be the case that simply identifying or highlighting users who have made a substantial contribution to the EB may be enough, since this has been shown to encourage others to do likewise. The number of reusers for a specific experience package, together with a record of how these reusers rated it could provide the basis for implementing a relatively simple and soft reward system.

When considering the initial stages of an EbIS it may be useful to consider the approaches taken for launching other types of KM systems meant to support organizations to share knowledge (e.g., [Wenger, McDermott, and Snyder (02)] and [Waterson (05)]). These often take the form of guidelines, principles, or heuristics and can be adapted to fit the EbIS context. Examples of these adaptations include:

Provide content at the launch of the EbIS: Without content an EbIS is not useful. A list of already included experience packages can give a perspective on the purpose of using an experience base and could help to create (initially at least) passive use. Active use is more likely to occur when the utility of the EB was already proven.

Stage the roll-out of the EbIS and plan ahead: Another strategy mentioned by the literature on knowledge management systems, speaks about launching a pilot system having a minimum of required functionalities, and planning carefully for later evolution and development. Providing too many facilities or being too ambitious (e.g., expecting users to be active from the start) proves to be common and leads users to visit only once and then rarely return. Similarly, each phase of the EbIS development should be planned; this should cover questions such as what extra facilities could be added later. What do users require? What has proved to be less successful than expected?

Facilitating the use of the EbIS: Alongside some initial content, the EbIS may need someone to facilitate the use and encourage contributions. This role can be assigned to knowledge brokers. Facilitating the deployment of an EbIS requires a lot of effort, however it pays off in terms of establishing it [Hellström, Malmquist, and Mikaelsson 00]. Sometimes, people need help in formulating their searches in order to be successful. Additionally, the knowledge brokers are aware of the current content of the EB and know the people who contributed it. They can recommend not only the best way to search for previous experiences, but also the people who encountered similar problems and could act as experts.

Monitor and evaluate the content of the EB over time: Monitoring the activity within the EbIS means more than keeping track of usage statistics and profiles. It also means regularly asking what kind of problems users encounter, and working toward solving them. Listening to the members and not taking their views, or indeed in some cases their apparent silence, for granted, also helps to sustain activity and establish

trust over time. Knowledge brokers should "walk the halls", trying to find out what are the reasons for mistrust and refusal to use the EB. They should also give feedback about weblog posts selected for being included in the EB and about the criteria used to select them, encouraging more people to contribute repeatedly. Regular updates of experience packages similarly make the EB more interesting and increase the likelihood of occasional users becoming more active. For example, long time unused experience packages should be sent for review to the initial contributors and to the people who tried to re-use them in time, so that they could decide if the respective packages should be completely removed or only updated.

4 Empirical Studies

The purpose of our empirical studies was to evaluate whether informal knowledge sharing using weblogs and wikis could be a content producing alternative for information elements and new experiences, hence reducing the effort involved. These studies comprised the monitoring of ten external weblogs, and the fostering of a Content Management System (CMS) including weblog and wiki facilities in a company Intranet.

For the analysis of entries from already existing external weblogs, ten such weblogs focusing on Software Engineering, Knowledge Management and eLearning were selected. During a two month-period, their posts were monitored by using news aggregators. Posts containing meaningful information for software engineers were retained and re-structured in order to be included either in the EB or in the IEB. Three experts with different backgrounds were involved in this activity.

The case study directed at using weblogs and wikis for collecting experiences in a company intranet was also carefully planned and launched. The CMS selected for this study was TikiWiki (http://tikiwiki.org).

The employees were given an introduction to social software allowing rapid, flexible and "on-the-spot" acquisition of knowledge (wikis, weblogs). They were encouraged to start and maintain personal and group weblogs for a two-month period and to post there all the experiences they considered meaningful for their work. Five employees started personal weblogs, while nine other community and project weblogs were created [see Tab. 1]. For organizing and structuring work-related content, wiki pages were recommended.

The content of four of these weblogs was also monitored during the two months period. The participants were provided with coaching and on-line support. A weekly newsletter was issued in order to provide information on social software and to maintain their interest in knowledge sharing. Aiming at encouraging knowledge exchange, the monitoring experts also acted as knowledge brokers for connecting the participants who shared the same interests. Meaningful content that could be used for documenting and including a new experience in the EB, or adding it to the IEB, was identified.

Weblogs and posts statistics: 114 out of 238 posts coming from the ten external and four Intranet monitored weblogs were retained as serving for our purposes.

The retained posts were counted using a) the classification of Meder [Meder (00)], and b) a SE-specific classification. The summary showed that these 114 selected posts contained (a) 62 definitions, 90 descriptions, 170 references, and 41

examples. We also found the descriptions (b) of 5 products, 5 processes, 12 techniques and 14 tools.

	No of users	**No of wiki pages**	**No of weblogs**	**No of posts**
After 1 month	26	66	13	181
After 2 months	26	72	14	215

Table 1. TikiWiki usage statistics

We illustrate these findings with two examples. The first example is about a lesson learnt that was included in the EB: A specific approach description for Test-First Programming was located in the *Mistaeks I Hav Made* weblog (http://nat.truemesh.com/archives/cat_testfirst_programming.html) written by Nat Pryce. "The problem was that I had written tests for each method, testing pre- and post-conditions and class invariants. This is the wrong approach to writing programmer tests. Instead, each test case should specify a describable aspect of the functionality an object provides to its clients. That aspect will probably involve multiple methods of the class. As a maintenance programmer using the tests as documentation you want to know how the behavior of those methods is interrelated, not how each method acts individually."

This post had to be re-structured in order to be included in the EB [see Fig. 2].

Subject	**Test-First Programming**
ID	1234
Topic Areas	SW-Technologies
Validity	0
Origin	#Nat Pryce (http://nat.truemesh.com/archives/cat_testfirst_programming.html)
Problem	The problem was that I had written tests for each method, testing pre- and post-conditions and class invariants.
Cause	
Situation	This is the wrong approach to writing programmer tests.
Reaction	Instead, each test case should specify a describable aspect of the functionality an object provides to its clients. That aspect will probably involve multiple methods of the class. As a maintenance programmer using the tests as documentation you want to know how the behaviour of those methods is interrelated, not how each method acts individually.
Prevention	
Current reusers	
Previous reusers	
Log	

Figure 2: Experience extracted from weblog post and prepared for inclusion in the experience base

The post also contained a reference to the definition of Test-Driven Programming from the C2 wiki (http://c2.com/cgi/wiki), dedicated mostly to eXtreme Programming. The definition was also re-used as a learning element [see Fig. 3].

Test Driven Programming
Extreme Programming supports the use of tests as a development tool. Given an object, the developers devise tests for all interesting methods even before programming them. This means that:
- they are forced to define precisely what a method does
- they know where to begin writing a method
- they know when you are done writing a method
- they know the minimal scaffolding needed to run a method
- like scientists, they target reproducible results .

This means that they will recognize dependencies among the objects early, and they will work to minimize them.

Figure 3: Learning element extracted from C2 wiki

The second example is about some learning elements identified in a post from *Bliki* (http://martinfowler.com/bliki/DomainSpecificLanguage.html), *Martin Fowler's weblog & wiki*. It included a description of domain specific programming languages, directly reused as a learning element. Moreover, we were able to extract from the post some pros and cons for this type of languages to create another learning element. Certainly the extraction cannot be done in an automatic way, but weblogs seems to motivate people to make their knowledge explicit, which can serve as starting point for knowledge elicitation. The extracted learning elements could be part of a "micro-didactical arrangement" supporting the comprehension of an experience that discusses the choice of a programming language for a certain project or that reports a lesson learnt of using a certain language for a specific domain.

Semi-structured interviews organized after the two months period obtained feedback from eight users and included 19 questions grouped in five categories (content, learning and motivation-related problems, usability, and possible future developments). The interviews took on average 15 minutes. The results are summarized here:

The TikiWiki environment had two categories of users: seven active contributors and 19 passive readers (lurkers). Since the content was generated without following any guidelines, it was extremely heterogeneous containing experiences, success and failure descriptions, research ideas, hints, tips and tricks, various links, references and comments. Some other content, related or not to Software Engineering, was also generated: conference announcements, hints for current and future PhD students, food recipes etc. Because the TikiWiki environment was only accessible from inside the company, there was no concern regarding trust or intellectual property problems. Contributors were more motivated by the reuse of knowledge for personal advantage than by sharing or adding new experiences to the experience base. A possible automation of knowledge discovery was also discussed. The opinions ranged from “definitely not possible” to “partial automation could be possible, but the decision on

including a post, or a page content partially or as a whole in the EB or IEB should be made by humans".

While only weblogs were specifically monitored during our study, they frequently contained references to wiki pages. Usually, weblog posts contain references, but also definitions, theories, strategies that are easier to retrieve when they are afterwards organized in wiki pages.

5 Conclusions and Future Work

In summary, we identified several problems of EbISs arising from a bias put on technical issues and disregard, (in some cases even ignorance) of human issues in socio-technical systems [Clegg 00]. A system that is designed to manage, retrieve, and present experience data, but does not fulfill the organizational, social, and psychological requirements to support knowledge sharing and learning will not get accepted by the users and finally, due to lack of input, its results will become less and less useful till it will be abandoned.

We introduced an approach meant to both provide the content to the experience base and improve learning from experiences within information spaces: weblogs serve as input for both an experience base and for an information element base; a pedagogical information agent retrieves suitable experiences to be further enriched with additional information elements and produce micro-didactical learning arrangements. Several guidelines, principles, and heuristics focussing on user motivating user and increasing trust were presented in this paper. Many of them have been applied during our empirical study. The study demonstrates that using weblogs for recording knowledge is a feasible alternative for supplying new experiences to be added to the EB and to be used in providing content for producing micro-didactical arrangements.

Besides the positive findings of our study, one important issue remains to be solved: The activities of documenting experiences (e.g., by writing weblogs or wikis), using EbIS, and learning within information spaces are still to isolated from the working tasks that are performed by a software engineer. The activities are still additional or complementary to the real working tasks. Therefore, in the future we intend to focus on the development of methodologies and technologies to support a closer integration of knowledge sharing tasks and learning tasks with the working tasks. Hence we will especially address the following research questions:

- Improving Knowledge Elicitation: How can we integrate the principle of open and informal documentation of experiences (e.g., by wikis or weblogs) into a concrete working task?
- Making learning more situated: How can we ensure that the created information space suits the current situation of the software engineer (e.g., current working task, problems to be solved, knowledge level and learning preferences of the engineer related to the working task)?
- Enabling Knowledge Seeding: How can we extend the information space so that additional Software Engineering knowledge is presented to the engineer compared to the knowledge he was explicitly looking for (i.e., striving to a

more *push-oriented* knowledge delivery instead of a *pull-oriented* knowledge searching)?

In order to improve knowledge sharing between software engineers and making learning more situated within the working environment, peer-to-peer and agent technologies will play a crucial role. To apply these technologies, we need more knowledge about how to describe working tasks and their surrounding environments formally, how to observe these tasks and their progress, and which impact delivered knowledge could have on a working task and its situation.

Acknowledgements

Dr. Gabriela Avram's work was carried out during the tenure of an ERCIM fellowship.

References

[Althoff and Pfahl (03)] Althoff K-D., Pfahl D.: "Making Software Engineering Competence Development Sustained through Systematic Experience Management"; in Managing Software Engineering Knowledge, A. Aybüke, R. Jeffrey, C. Wohlin, M. Handzic (Eds.), Springer, Heidelberg / New York (2003)

[Angeles 03] Angeles M.: "Making Sense of Weblogs in the Intranet"; presentation at Usability Professionals Association, 16 September 2003, http://studioid.com, (retrieved on 2005/01/31)

[Clegg 00] Clegg CW.: "Sociotechnical Principles for System Design"; Applied Ergonomics, 31, (2000), 463-477.

[Coates 03] Coates T.: "My Working Definition of Social Software", Plasticbag, http://www.plasticbag.org/archives/2003/05/my_working_definition_of_social_software.shtml (retrieved on 2005/01/31)

[Ericsson, Krampe, and Tesch-Römer 93] Ericsson K. A., Krampe R. T., Tesch-Römer C.: "The Role of Deliberate Practice in the Acquisition of Expert Performance"; Psychological review, 100, (1993), 363-406.

[Hall 01] Hall H.: "Input-friendliness: Motivating Knowledge Sharing across Intranets"; Journal of Information Science, 27 (3), (2001), 139-146.

[Hellström, Malmquist, and Mikaelsson 00] Hellström T., Malmquist U., Mikaelsson J.: "Descentralizing Knowledge – Managing Knowledge Work in a Software Engineering Firm"; VRR-00-26, Journal of High Technology Management Research, 2, 3 (2001)

[Klusch 01] Klusch M.: "Information Agent Technology for the Internet: a Survey"; Data & Knowledge Engineering Archive, 36, 3 (2001), 337-372.

[Kolb (84)] Kolb D. A.: "Experiential Learning: Experiences as the Source of Learning and Development"; Prentice Hall, New Jersey (1984)

[Marsick and Watkins 02] Marsick V., Watkins K. E.: "Informal and Incidental Learning"; Merriam, S. B. (ed.), The New Update on Adult Learning Theory (New Directions for Adult and Continuing Education), no. 89, Jossey-Bass, (2001), 25-34.

[Meder (00)] Meder N.: "Didaktische Ontologien"; Ohly H. P., in Rahmstorf G., Sigel A. (Eds), Globalisierung und Wissensorganisation: Neue Aspekte für Wissen, Wissenschaft und

Informationssysteme., 6: Fortschritte in der Wissensorganisation, Ergon Verlag, Würzburg, (2000)

[O'Dell 98] O'Dell C., Grayson C. J.: "If only We Knew what We Know: Identification and Transfer of Internal Best Practices"; California Management Review, 40, 3 (1998), 154-174.

[Ras and Weibelzahl 04] Ras E., Weibelzahl S.: "Embedding Experiences in Micro-didactical Arrangements"; Proc. 6th International Workshop on Advances in Learning Software Organisations, Banff, Canada, (2004)

[Rech, Ras, and Jedlitschka 04] Rech J., Ras E., Jedlitschka A.: "Improving Software Quality through Refactoring by means of Didactical Augmented Experience"; Beydeda S., Gruhn V., Mayer J. D., Reussner R., and Schweiggert F., (Eds.), Testing of Component-Based Systems and Software Quality, vol. P-58, Lecture Notes in Informatics (LNI), (2004), 141-149.

[Rus and Lindvall 02] Rus I., Lindvall M.: "Knowledge Management in Software Engineering"; IEEE Software, (May/June 2002), 26-38.

[Stenmark 01] Stenmark D.: "The Relationship between Information and Knowledge"; presented at *IRIS-24*, Ulvik, Norway, August 11-14, (2001), http://w3.informatik.gu.se/~dixi/publ/iris24-DS.pdf (retrieved on 2005/01/31)

[Stenmark 02] Stenmark D: "Intranet-supported Knowledge Creation: Factors and Technology for Organisational Creativity"; Proc. of OKLC 2002, Athens, Greece, (2002), http://www.viktoria.se/results/result_files/185.pdf (retrieved on 2005/01/31)

[Tautz (00)] Tautz C.: "Customizing Software Engineering Experience Management Systems to Organizational Needs"; PhD thesis, Technical University of Kaiserslautern, Germany, Fraunhofer IRB Verlag, Germany, (2000)

[Waterson (05)] Waterson P. E.: "Motivation in On-Line Communities"; Dasgupta S. (ed.), to appear in Encyclopedia of Virtual Communities (ed.), George Washington University, (2005)

[Wenger, McDermott, and Snyder (02)] Wenger E., McDermott R., Snyder W.M.: "Cultivating Communities of Practice"; Boston: Harvard University Press, (2002)

[Wikipedia 05] Wikipedia, the Free Encyclopedia , http://en.wikipedia.org/wiki/, (retrieved on 2005/01/31)