

Using Knowledge Discovery Technology in Experience Management Systems

Jörg Rech, Björn Decker, Klaus-Dieter Althoff

Fraunhofer Institute for Experimental Software Engineering (IESE)
Sauerwiesen 6
67661 Kaiserslautern, Germany
{rech, decker, althoff}@iese.fhg.de

Abstract: The management of experiences from projects is an important issue for modern software organizations. In this paper we describe our initiative for experience management (EM) called COIN based on case-based reasoning technology. We present several examples of application from data mining and text mining to enhance our system at Fraunhofer IESE. Our goals are, on one hand, to discover new knowledge and, on the other hand, to use this knowledge to improve the processes within an EM organization.

Keywords: Experience Management, Experience Factory, Case-Based Reasoning, Software Engineering, Text Mining, Data Mining, KDD.

1. Introduction

Knowledge as the fourth factor of production is one of the most important assets for any kind of organization, and for all areas of science. While *experiences* describe events in one specific context that can only be used carefully, *knowledge* is usually applicable in previously unknown contexts with a fair amount of certainty. Unfortunately, the knowledge of an organization is mainly held by few experts who acquired it through their experiences in day-to-day work or research. With the fast growth of our institute (Fraunhofer IESE), we faced the problem that the more beginners we had the more communication was needed to transfer knowledge from the experts to the newbies. To master their tasks they needed knowledge that was not taught at the university. Either the newbies had to access their own limited experiences or learn from colleagues by personal communication. But our experts were already rarely accessible because of their involvement in many different tasks. Nevertheless, it is important to provide newbies with default guidelines and facilitate experience sharing among them.

The research area dealing with support for “learning from experience” is called *Experience Management* (EM) ([Kluge 1999], [Bergmann 2001], [Tautz 2000]). It deals with the development of methods for the identification, storage, improvement, propagation, and re-/use of experiences from members of an organization. One goal is to enable non-experts to access relevant guidelines and a higher number of alternative decisions from all sources of knowledge in order to effectively and efficiently master current tasks. Software engineering (SE) as an application area of EM depends

heavily upon the experience of experts for the development, application and advancement of its methods, tools and techniques. From this, an approach for EM was developed in the mid-eighties, which is called *experience factory* (EF) [BCR 1994].

Since the size of our institute does not allow to talk to experts on a frequent basis, we launched an internal project *COIN* (COrporate Information Network) based on the EF idea to foster our EM initiative. The technical infrastructure of COIN is based upon *case-based reasoning* (CBR) technology [Kolodner 1993]. In our effort to optimize and enhance our current system, we are planning to include technology from the *Knowledge Discovery in Databases* (KDD) field. The discovery of hidden and previously unknown knowledge from data is the goal of the data mining (DM) [ES 2000] step in the KDD process. We are pursuing the same goal for the experiences stored in COIN. Since experiences in an EF are structured (hyper-)textual documents, we are more interested in text mining (TM) and web mining (WM) techniques.

In this paper we first give an overview of our research project — the implementation of an EF called COIN and describe open work and research opportunities. Thereafter, various approaches for the usage of KDD techniques within an EF are explained. We continue with a description of the next steps in our project and close with a short summary.

2. Fraunhofer IESE’s Corporate Information Network (COIN)

COIN was launched to foster our EM initiative and emerged from an experience base for CBR systems

called CBR-PEB [ANT 1999]. Beside the propagation of knowledge within our institute, COIN is used as a real-world environment for the development and validation of technologies and methods for goal-oriented EM. This includes knowledge identification, collection, processing, propagation, presentation, and maintenance, as well as experience evaluation, analysis, generalization, specialization, and formalization. The COIN environment consists of three main parts: the experience base (EB), the COIN team, and an intranet-based interface.

Within the EB of COIN, all kinds of experience necessary for our daily business are stored (e.g., business processes, guidelines, or observations). Defined *processes* (structured interviews; see Fig. 1) populate this EB systematically with experience typically needed by our project teams [TAN 2000].

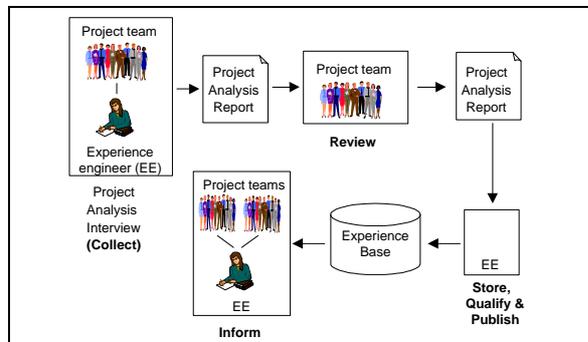


Fig. 1: Project analyses in COIN

From the project analysis report (PAR) we extract useful experiences. New or reused experiences are processed based on our packaging procedure [ABT 1997] that was repeatedly refined [Tautz 2000]. Dedicated improvement processes analyze problems that have occurred, devise improvement actions to avoid their recurrence, and implement strategic decisions by the institute’s leadership. Project teams using process descriptions and gaining experiences cannot be expected to invest the effort to manage these experiences. Compared to the objectives of the organization, projects have a short-term perspective, focusing on the development goals of the project. Therefore, an *organizational unit*, that is responsible for experience management is required and has to be separated from the project teams. According to [BCR 1994], this separate organizational unit is called EF, which for the IESE is operationalized by the *COIN team*.

Common requirements for an EB are to support different kinds of interrelated experiences and the need for context-sensitive, similarity-based retrieval [Tautz 2000]. This demands a specialized *technical infrastructure* for the EB. Our solution to these requirements is called *INTERESTS* (INTElligent RETrieval and STORAGE System) [ABH et al. 1999]. As shown in Fig. 2 INTERESTS consists of a tool layer

for accessing and presenting the EB contents using a standard web browser, a general purpose EB server, and a commercial CBR tool (formerly *CBR-Works* but now *orange* from tec:inno, Germany), which is used for the actual EB.

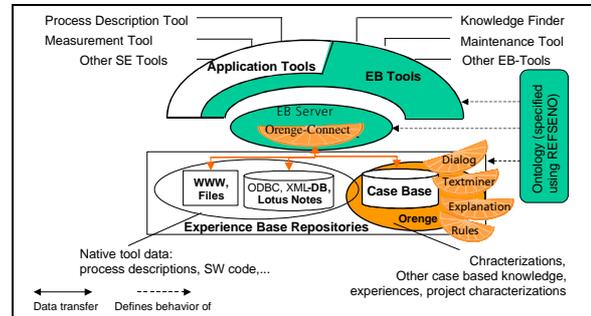


Fig. 2: COIN’s technical infrastructure (INTERESTS)

Within an experiment the benefits of this EB approach have already been demonstrated [Tautz 2000]. Until now we have gathered nearly two years of operational experience in maintaining COIN, and we have successfully adapted COIN to partners/customers, for example in the *IPQM project* for continuous improvement of hospitals in the German healthcare sector [ABMN 1999]. Based on these experiences, we have widened the requirements of COIN towards an organization-wide information and knowledge management system.

The Experiences in COIN

Since we started COIN in 1999 with about 250 experiences, it has grown to 550 experiences. We expect a continuing annual growth of an equal amount for the coming years. In addition to the sheer amount, experiences are highly interrelated and context-sensitive. For example, observations and problems are gained during a project while a particular business process was performed. Such experiences are unique in the sense that the same context will not recur. People will have to search for experiences that have been gained in similar contexts and adapt it to their own.

To support the retrieval of the experience in COIN, each is implemented as a “case” based on a structural CBR approach. A domain *ontology* is used for modeling the different types of case concepts, formal and informal attributes together with the respective similarity measures, as well as relations between cases. Our ontology is based on REFSENSO (REpresentation Formalism for Software ENgineering Ontologies) [TG 1998], which is tailored to our storage and retrieval needs.

Experiences of COIN originate in projects and are recorded in the project analysis phase with structured interviews (see Fig. 1). As shown in Fig. 3, the project

analysis report (PAR) is also a valuable resource for knowledge about projects.

Question 8: What did you learn about the customer?
A-LL: A likes to do as much as possible on their own, even if available resources are insufficient.
A-GQM:
 •A wants to change as little as possible.
 •It is important to agree to a customer's wishes.
A-LL and A-GQM:
 •Even if those A employees involved in a project react very positive within meetings, it can happen that IESE suggestions are not seized because of the unavailability of the required resources/money to be provided by the responsible manager.
 •IESE project members have to learn to differentiate between customer employees being present during meetings and the responsible managers who have to provide the required resources/money.
 •IESE project members have to learn to talk to their customers »close to a possible offer« and in such a way that it is realistic for the customer.

Fig. 3: Extract of a project analysis report

The experiences stored in COIN are structured text documents centered around two major subject areas: business process models and lessons learned. As shown in Fig. 4, an experience consists of various parts, such as a description or the project objective, together with metadata like the experience type and category.

Project Experience "Less Effort" (ID 2183)

Type: Observation
Category: Best practice experience

Description: "Although project was negotiated to end on Sep 30, 1999, the project work was already finished on July 15, 1999. The reason was that the systems we were to measure were provided earlier than expected. Thus, analysis could start and finish earlier."

Objective of project:
 "Investigation of the impact of distribution techniques and programming languages on the maintainability and reusability of software systems for space applications."

Funding: Industrial
Project type: R&D
Project manager: Jürgen W.

Fig. 4: Shortened example of a project experience

The *lessons learned* (LL) can cover different topics and take on different forms [BT 1998]. Within COIN, LL about project management are captured. One LL can take on the form of an *observation*, a *problem*, *guideline*, *pragmatic solution*, or an *improvement suggestion*. Each LL is personalized to allow a querying IESE member to ask a colleague for further information. The context of these LL are modeled by the two concepts "project" and "process". A "project" is a characterization of the project in which the lesson learned was gained (e.g., person month, duration). The "process" concept names the business process and thus the project phase in which the LL was gained. Therefore, a project team member can specify his current environment as well as the current situation to search the EB for similar experiences. Fig. 5 shows the

interrelations between the context and the different types of LL.

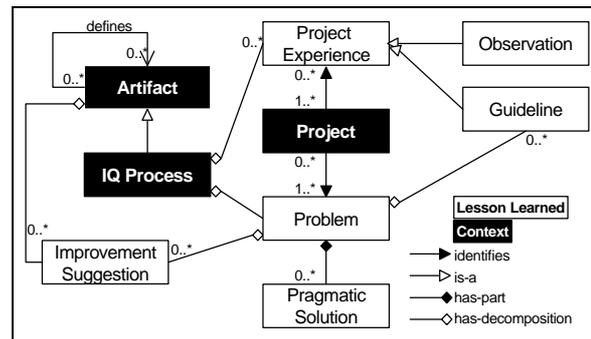


Fig. 5: COIN Ontology according to [Tautz 2000]

Observations are facts that are of interest to future projects, often expressing some baseline (e.g., "it took 10% of the total effort to manage the project") or some positive effect (e.g., "the customer was happy because we provided him with a ready-to-use tutorial"). *Problems* are descriptions of negative situations that occurred during a project (e.g., "the expectations of the customer were not met"). *Guidelines*, *improvement suggestions*, and *pragmatic solutions* relate to one or more problems. *Guidelines* are recommendations on how a particular business process should be performed. For example, a guideline could be the following: "Interact with the customer frequently, at least twice a month." An *improvement suggestion* is a proposal to change an artifact to avoid problems that occurred during its usage. *Pragmatic solutions* are sequences of immediate countermeasures taken by a project team in response to a recognized problem. While a guideline aims at preventing a problem from occurring in the first place, a pragmatic solution is applied after a problem has already occurred.

Business process models represent procedural knowledge and are mostly prescriptive. That means they are formulated as instructions for process execution. While process models like best-practices are to be seen as recommendations, some, such as administrative procedures, have to be followed without exception.

Recently, an extension of our ontology was realized with a visiting scientist [Dingsøyr 2001]. We enhanced the ontology by adding seven categories and classified the existing experiences. The categories had the following focus:

- *Best Practice experience* are work practices that can serve as an example for others.
- *Risk experience* describe factors that might influence the execution and result of a project in a negative way. It can be enriched with suggestions for prevention and emergency counteractions.

- *Technology experience* are impressions of how a technology has been applied in practice, and problems or successes that have arisen.
- *Customer experience* describe the impressions of the customer in a given project. This can include information about the likes and dislikes of contact persons or the internal organization, and culture within the customer’s organization.
- *Product models* show the relationships between variables in processes or projects, such as effort spent on different tasks in a project, and which influence that factor has on project duration.
- *Research experience* is related to research goals or to the scientific content of an industrial project.
- All remaining experiences were grouped under the *miscellaneous experience* category.

We classified the first 240 experiences from the three classes guidelines, observations, and problems into the extended ontology. Table 1 shows the distribution regarding the new categories (Experiences can be classified into multiple categories).

	Guideline	Observation	Problem
Best practice	41	27	44
Risk	0	3	9
Technology	2	5	5
Customer	25	44	20
Product	1	1	0
Research	0	0	11
Misc.	4	7	26

Table 1: Reclassification of COIN experiences

As shown in Fig. 6, most of our experiences are accumulated in the categories “best practice” and “customer experience”.

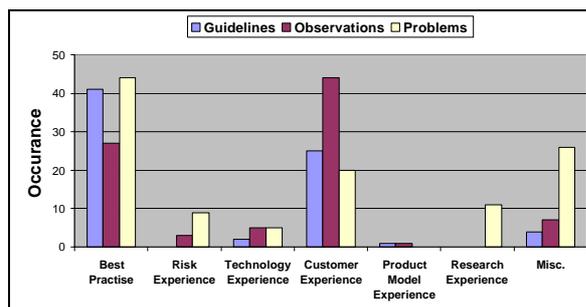


Fig. 6: Distribution of experiences in COIN

During classification of the experiences, we had to group “worst practices” into the same category as “best practices”. Table 2 illustrates how many experiences (guidelines/observations/problems) were classified into multiple categories. For example, ten guidelines from the “best practice” category were also put into the

“customer” category and five problems into the “miscellaneous” category.

	Customer	Product Model	Research	Misc.
Best practice	10/4/1	1/1/0	0/0/2	0/1/5
Risk	0/1/1	0/0/0	0/0/0	0/0/0
Technology	0/1/1	0/0/0	0/0/1	0/0/1
Customer	—	0/0/0	0/0/0	0/0/0

Table 2: Multiple classification in COIN

Based on the mentioned categorization and classification results we draw the following conclusions:

- Only few experiences are focused on technology, risk management or product models. These are topics where we need to invest further energy to accumulate helpful experiences.
- We already have collected a large number of “customer” and “best practice” experiences. A potential user of these experiences could be frustrated by this flood of documents. Here we should split the categories and define an ontology with a finer resolution to support experience retrieval. For example, we either have to rename “best practices” to “process experience” or split it into two different categories (i.e., “worst” and “best” practices).
- We also discovered that many experiences deal with problems related to the scientific outcome of projects in the “miscellaneous” category. From this we might introduce a new category to reduce the number of experiences.

Current Work in Progress

As mentioned in past publications (e.g., [ABMN 1999]), we want to include technology for data mining into COIN to extract additional and previously unknown knowledge from an EF.

Several questions arose from this. How do we start and guide the information flow? Is it possible to minimize the information flow so we do not drown the user in it? Can we generalize several similar experiences to obtain comprehensible and customizable knowledge? Can we support the definition of goals, questions, and metrics from already existing experiences? Can we derive additional questions or goals from the required metrics respectively their collection processes? Can we support the construction or evolution of an ontology with knowledge from already existing experiences? Can we determine, maintain and improve the “quality of experiences” with methods from data mining? How should we handle the knowledge from the EF usage to support other users?

In the following chapters we have conceptualized several approaches in more detail to answer these questions.

3. Knowledge Discovery in COIN

The techniques of data mining or text mining are valuable tools for the discovery of previously unknown knowledge. This knowledge represents new facts that can either be used in new software projects or in the improvement, management, or support of processes within the EF organization. We are mainly interested in the following groups of techniques:

- *Classification* techniques enable the categorization of experiences into previously defined groups of experiences and provide information on the experiences within a group.
- *Clustering* helps to find groups (i.e., clusters) of experiences and information about the center or the rim of such a group. It can be used to detect deviations from the core cluster — e.g. either extremely good or bad practices.
- *Association rules* are used to detect trends over time or to find information about relations between experiences, for example, similar experiences extracted by a specific group of users.

To improve COIN, we are in the process of developing methods to discover new knowledge in experiences, to detect, maintain and improve the “quality of knowledge”, to support the construction and evolution of EF’s, and to support COIN users.

We now describe various approaches to solve the problems of the previously mentioned research topics based on text mining and data mining.

Experience Generalization and Aggregation

Experiences describe events, problems, and solutions in a particular context (e.g., a specific project). The tailoring of experiences in other contexts is a time-consuming process. A promising task is therefore the automatic extraction of valid and significant knowledge that is applicable in new contexts.

Generalization of experiences denotes the extraction of underlying rules and laws from experiences. As depicted in Fig. 7, the idea is to find a commonality in various experiences (e.g., experience A, B, and C) and to extract the “general” knowledge (G) or rule. *Aggregation* means the packaging of various experiences (e.g., experience A, B, and C) to create a new one, for

example, a project plan (A) with an appropriate test (B) and quality plan (C).

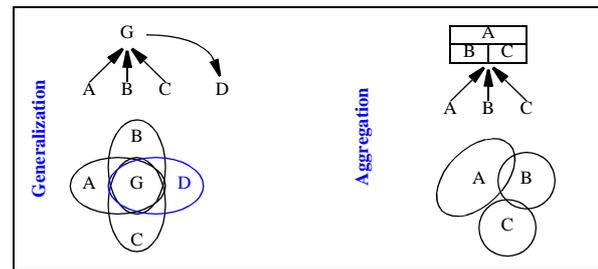


Fig. 7: Generalization and aggregation

Our idea is based on the discovery of commonalities and variabilities between experiences in an EB. This can be done either by using the existing characterization and using all experiences of one class or by the construction of profiles (see the next section). Thereby we want to collect very similar experiences or parts thereof and minimize the amount of documents that are important candidates for the generalization process.

Beside the stored experiences within COIN, we have another pool of experiences to use. We are currently running a project in which project members can discuss ongoing tasks. This can either be used to find agreement on a specific topic or to find and discuss the best solution for a given problem within a larger community. Out of this we are trying to summarize the long, often contradictory and always unstructured discussions to extract knowledge that can be integrated into COIN.

Characterization of New Experiences

The flood of experiences from real projects into the EF increases as the EF is more established and used. Automatic processes for the discovery, capturing, dissemination and characterization of large amounts of experiences are needed to speed and ease the experience acquisition process. We understand *characterization* as the determination of attributes, values, and relations of an experience item based on the given ontology. We do not mean the *classification* of already characterized data into a new class hierarchy or the clustering of already described experiences into formerly unknown groups of experiences (cf. Fig. 8).

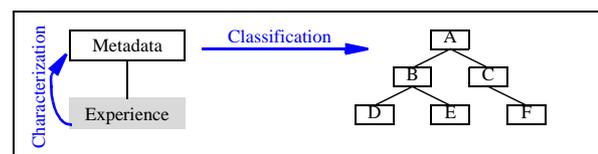


Fig. 8: Characterization and classification of Data

To characterize an experience the computer has to understand what the experience is about. As this approach is still very complex, we try to characterize new experiences by extraction of meaningful words.

For example, we want to know if an experience is a “customer experience”. Based on the previously known industrial partners we can extract the funding partner from the PAR (cf. Fig. 1) or the structured text of the experience. Because we only have “customer experience” with industrial partners, academic or in-house partners are not considered.

Another approach is to characterize new experiences by constructing profiles of their words, sentences and paragraphs with techniques from Text Mining. Parts of the profile can be weighted based on their position in the structured text or based on domain specific knowledge (e.g., terminology from software engineering). These profiles are then compared to already characterized experiences within the EB and point at probable values for the classification.

Because the characterization is a unguided procedure it is prone to errors. There are two ways to handle the resulting metadata. Either it is immediately verified by a human expert in a “normal” characterization process or the experience is evaluated and re-characterized by the users of COIN. In the latter approach, these experiences have to be marked and interlaced into unsimilar queries to give them a chance even if their characterization is defective.

Currently, we are looking for expressive patterns in the experiences for characterization and classification techniques. Subsequently, we have to extend our technical infrastructure, and define methods for the evaluation and correction of characterization results.

Maintenance of Experiences

The quality of the experience is the most important aspect of an EF. As time goes by, even the best experience ages and gets out-of-date or it is never reused because of an incomprehensible content. This can be caused by changes in the organization, the industry, or by scientific innovations. More straightforward causes are mischaracterization or misdescription of experiences. To maintain and restore the “quality of knowledge” we need to detect old, inconsistent, or incomprehensible experiences.

Our current approach is a manual feedback loop from the EF users to its maintainers as depicted in Fig. 9.

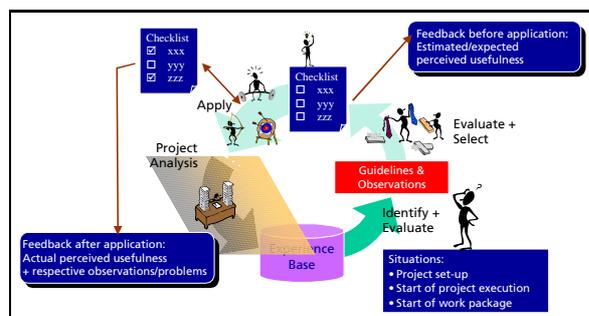


Fig. 9: User feedback in COIN

This feedback indicates if the experience is useful, not useful, or irrelevant for the user. This process can be supported by techniques from data mining. Analog to the characterization of experiences, we have to compare new experiences with old ones to detect inconsistent or contradictory experiences. After we find similar experiences, we either have to mark them for maintenance or generalize them as in a previous approach. In addition to the user feedback, frequency of usage, and validity ([NA 2001], [NAT 2001]) we can find experiences similar to “incomprehensible” ones and mark them.

Evolution of Ontologies

Another way to use DM in the context of an EF is the support of the EF itself. Typical events in the life of an ontology are its extension, and its fusion with other ontologies. There are different reasons for these events:

- As the EB of an EF grows, some classes of the ontology accumulate *too many experiences*.
- It can happen that external influences or internal decisions cause the *change of goals* and reuse scenarios in an EF. Either new goals are formulated, which have to be put into practice, or old goals are modified or removed.

The change of goals would result in an EF construction process with DISER and a reclassification of existing experiences with the new ontology. Based on the information (characterization or word profile) in the structured text or the project analysis report (see one of the previous sections), a new characterization based on the given ontology has to be determined.

In case we accumulated too many experiences the classes either have to be refined or the amount of experiences has to be reduced. In our case the refinement of our ontology would also include the reformulation of goals and reuse scenarios that were defined in the construction process. As mentioned earlier we already extended our ontology with additional classes. We noticed, for example, the large “best practices” category (cf. Fig. 6), which could result in being split into a “worst practices” and a “best practices” category. As we do not have a proven process to reformulate the reuse scenarios and goals, we are currently evaluating possible strategies and their effects.

To support the generation of an ontology we can also use information within the experiences. The basic idea is the *re-engineering of an ontology*, reuse scenarios, and EF goals from the experiences alone. The ontology can be constructed by using clustering techniques that find hierarchically dependent classes of experiences to build a classification. This automatically developed ontology can be used to verify the validity of the currently designed ontology. Differences between the

ontologies have to be resolved. This can cause the refinement, merging, or transformation of classes, attributes, similarities and values of the present ontology. Goals and reuse scenarios have to be derived by comparing similar ontologies from past experiences with EF construction processes.

1. Construct the new *ontology* based on available experiences.
2. Enhance the constructed ontology by giving the unnamed classes meaningful names by reuse of the designed or other ontologies.
3. Compare constructed ontology with designed ontology to find classes with:
 - too much/few experiences
 - unused experiences.

Analyze these anomalies and store this information for adaption of the *record processes*.

4. Examine retrieval behavior of the EF users and store information for adaption of the *reuse scenarios*.
5. Examine the impact of new record processes and reuse scenarios on *goals* and objectives.
6. Redesign the EF ontology with DISER.

Another application of the previous approach is the determination of additional reuse scenarios and goals from experiences with the construction of similar EF's.

Similarity between Experiences

One primary quality of CBR is the similarity based retrieval of problem/solution cases. But how do we define similarity between two cases or — if we go deeper into the ontology — between values or attributes? By using the methods of text mining (TM) we are planning to determine similarity directly between experiences or values of the ontology.

The similarity between values of an ontology (e.g., between value “organization A” and “organization B”) can be determined with the similarity between the experiences with the given values. Let the similarity between different values be 0 (0%) except for the similarity of a value to itself which is 1 (100%). The procedure could be as follows:

1. Choose two experiences with the observed values that have as few different other values as possible.
2. Remove the observed attribute and determine the similarity between two experiences
 - either by calculating the frequency of equal words of both experiences.
 - or by adding the similarity between values in the characterization of both experiences.
3. Increase or decrease the similarity between the two values of the removed attribute, (e.g.,

decrease if less than 50% similar and increase if more than 50% similar).

For example, if two experiences have the same characterization except for the customer (i.e., organization), the similarity of the two organizations would increase. In later iterations of this procedure we could infuse information from the already determined similarity into the similarity calculation (step 2).

Experience Factory Usage Support

Beside supporting the construction of an EF, data mining can be used to analyze its usage.

With techniques from “clickstream analysis” we can determine the retrieval behavior of COIN users to minimize retrieval sessions or to customize access for every user. Information about retrieval behavior can expose reuse scenarios and record processes that were not included in the original EF construction. Based on this new information the maintenance of the EF can be triggered and the EF design can be tailored better to the users.

With information about the needs of the users we can generate user profiles. These can be used to find groups of people with similar interests within an organization or to “push” new information to the user as soon as it is available. This makes it possible that a newbie can profit from an EF right from the start.

4. Summary and Outlook

In this paper we presented the applicability of knowledge discovery technology in the context of the experience management (EM) initiative at Fraunhofer IESE. After a brief introduction we described our corporate information network (COIN) for EM that is based on the experience factory (EF) approach. We described two sources for knowledge discovery, namely the structured text of the experiences and the project analysis reports. Following this, we presented our ontology and experiences with the extension and reclassification of existing experiences.

In the core of our paper, we described various research topics regarding COIN. Our approaches to tackle these are based on methods from data mining (DM) and text mining (TM). We described our plans to develop methods to discover new knowledge in experiences, to detect, maintain, and improve the “quality of knowledge”, to support the construction and evolution of EF's, and to support COIN users.

One commonality of the mentioned approaches is the computation of similarity between two different experiences without the usage of their characterization (metadata). As a next step we therefore plan to evaluate different systems that analyze text documents to extract metadata for later comparison.

We are also evaluating some classification tools based on decision trees and support vector machines (SVM's). On the one hand, we hope to discover previously unknown knowledge about our experiences and on the other hand, we want to define a process to semi-automatically classify experiences into COIN.

5. References

- [ABH et al. 1999] Klaus-Dieter Althoff; Andreas Birk, Susanne Hartkopf; Wolfgang Müller; Markus Nick; Dagmar Surmann; Carsten Tautz. *Managing software engineering experience for comprehensive reuse*. In Proc. of the 11th International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Kaiserslautern, Germany, June 1999.
- [ABMN 1999] Althoff, K.-D., Bomarius, F., Müller, W. & Nick, M. *Using Case-Based Reasoning for Supporting Continuous Improvement Processes in Hospitals*. In Proc. of the 10th German Workshop on Machine Learning (FGML 1999), Universität Magdeburg, 27.9.-1.10. 1999.
- [ABT 1997] Althoff, K.-D., Birk, A. & Tautz, C. *The Experience Factory Approach: Realizing Learning from Experience in Software Development Organizations*. In: Proc. of the 10th German Workshop on Machine Learning (FGML 1997), Universität Karlsruhe, 6.-8.8.1997.
- [ANT 1999] Althoff, K.-D., Nick, M. & Tautz, C. *CBR-PEB: A Tool for Implementing Reuse Concepts of the Experience Factory for CBR-System Development*. In Proc. of the 7th German Conference on Knowledge-Based Systems (XPS 1999) Workshop on Case-Based Reasoning (GWCBR 1999).
- [Bergmann 2001] Bergmann, R. *Experience management - foundations, development methodology, and internet-based applications*. Postdoctoral thesis, Department of Computer Science, University of Kaiserslautern (submitted).
- [BCR 1994] Basili, V.R.; Caldiera, G.; Rombach, D. *Experience Factory*. In Marciniak, J.J. (eds.), "Encyclopedia of Software Engineering", Vol. 1, 469–476; John Wiley & Sons, 1994.
- [BT 1998] Birk, A.; Tautz, C. *Knowledge Management of Software Engineering Lessons Learned*. In Proc. of the 10th Conference on Software Engineering and Knowledge Engineering (SEKE'98), San Francisco Bay; USA. 1998.
- [Dingsøyr 2001] Dingsøyr, Thorgeir: *Classification of collected experience in CoIN*. Internal Report, Fraunhofer IESE, 22.3.2001
- [ES 2000] Ester, M.; Sander, J. *Knowledge Discovery in Databases: Techniken und Anwendungen*. Berlin: Springer Verlag, 2000, ISBN: 3540673288.
- [Kluge 1999] Kluge, A. *Erfahrungsmanagement in lernenden Organisationen*. Göttingen: Verlag für angewandte Psychologie, 1999.
- [Kolodner 1993] Kolodner, J.L. *Case-Based Reasoning*. Morgan Kaufmann Publishers, 1993
- [NA 2001] Nick, M., and Althoff, K.-D. *Engineering experience base maintenance knowledge*. Technical Report, IESE-Report No. 018.01/E, Fraunhofer IESE, Kaiserslautern, Germany, 2001.
- [NAT 2001] Markus Nick, Klaus-Dieter Althoff, and Carsten Tautz: *Systematic Maintenance of Corporate Experience Repositories*. In David B. Leake, Barry Smyth, David Wilson, and Qiang Yang (eds.) "Computational Intelligence Journal - Special Issue on Maintaining Case-Based Reasoning Systems", Volume 17(2), pages 364-386. Blackwell Publishers, 2001.
- [Tautz 2000] Tautz, Carsten: *Customizing Software Engineering Experience Management Systems of Organizational Needs*. Stuttgart: Fraunhofer IRB Verlag, PhD Dissertation, Dept. of Computer Science, University of Kaiserslautern, Germany, 2000.
- [TAN 2000] Tautz, Carsten; Althoff, Klaus-Dieter; Nick, Markus: *Learning from Experience — An Experience Factory Case Study*. In Proc. of the 13th German Workshop on Machine Learning (FGML 2000), Sankt Augustin, Germany, 2000.
- [TG 1998] Carsten Tautz; Christiane Gresse von Wangenheim: *REFSENO: A representation formalism for software engineering ontologies*. Technical Report, IESE-Report No. 015.98/E, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, 1998.