# Case Based Reasoning for Knowledge Management in KDD-Projects

## Concepts, Organizational Setting, Categorization into KM and Application in the case of Knowledge Discovery in Databases

Kai Bartlmae
DaimlerChrysler AG,
Research and Technology 3
D-89013 Ulm, Germany
kai.bartlmae@daimlerchrysler.com

Michael Riemenschneider
DaimlerChrysler AG,
Research and Technology 3
D-89013 Ulm, Germany

## Abstract

In this paper we introduce our departments organizational and technical infrastructure for knowledge-intensive and weak-structured processes: A framework for Knowledge Management in the case of projects in Knowledge Discovery in Databases (KDD). It is based on the experience factory approach and the method of case based reasoning. We introduce both approaches in the context of knowledge management, derive application-areas and introduce our realization for projects in knowledge discovery in databases.

## 1 Introduction

Many knowledge intensive activities take place in project organizations, where project teams form a temporal organization, which are disbanded after the projects are completed. This shows especially true for the work, our department FT3/AD is involved in, Knowledge Discovery in Databases. Here we analyze customer databases of different DaimlerChrysler branches i.e. for marketing reasons or for assessing credit risk. Because we work in these temporal teams, it is our interest that the experience gained in these projects should not only be kept as the team members personal knowledge, but be kept within our business organization in order to be reused.



**Proc. of the Third Int. Conf. on Practical Aspects of Knowledge Management (PAKM2000)**
**Basel, Switzerland, 30-31 Oct. 2000,** (U. Reimer, ed.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-34/

This is not a new problem, so i.e. Heisig [HEI98b] proposes that before a new project is being started, a plan for collecting know-how and experiences should be prepared, considering the topics of:

- Who is responsible for experience collection?
- Where can know-how be gained?
- Who gained certain experiences?
- In what form should the experience be documented?
- How are the experiences collected and saved?
- How are the experiences be disseminated?

But experience documentation has many barriers, so it is a time intensive task and the person documenting it will in many cases not be the user of it and therefore reluctant to share [KPMG98]. Further, the project teams are under time pressure and therefore the motivation for documenting experiences is initially low. A goal of an approach must be to give the team members help and time when documenting their own project experiences as well as giving them the information they need as easy and quick as possible, releasing them from administrative work. Further, project-management has to make the team aware of the need for knowledge management, to define processes for it, to train the teams, and last but not least to introduce a technical infrastructure to collect, disseminate and reuse them.

Here we try to approach these problems with the concept of case based reasoning together with the experience factory concept by Basili et al. [BCR94], building the base of the Experience Factory in Knowledge Discovery in Databases at FT3/AD (see also [BAR99]). It covers necessary aspects mentioned above for knowledge management in project work. The approach has its basis in the domain of software engineering and successfully be applied by Althoff et al. [ABT97].

In this paper we introduce both concepts, show where they can complete each other and how they cover the different building blocks of knowledge management. The paper concludes with the description of the KDD-experience factory, describing selected experience package types used.

## 2 Organizational view: The Experience Factory approach

The approach of the experience factory has been introduced by Basili et al. as an evolutionary, experience based approach for the improvement of software-products and software-development-processes. They were motivated by the realization, that collected experiences can improve development processes[HOU99]. Based on the Quality Improvement Paradigm (QIP), the experience factory has been introduced as an organization that supports the projects teams in the different steps of QIP.

One basic determinant of it is its organizational separation from the project teams in order to compensate the different goals project-teams and experience-management have[ABT97]:
While project teams try to reach their project goals fast and within a cost-frame, experience management wants the avoidance of mistakes or the installation of good practices using collected experiences. But this process of experience collection is time consuming and costly, meaning additionally effort for the project members. This is why an organizational separation of collection and the creation of experiences might prove useful. The organization for collecting, structuring, saving and disseminating of experiences is called an Experience Factory by Basili. Experience packages (EP) are its form for representing experiences of different structure and types, from data to process definitions. These are saved in an experience base, which can be compared to a safeguarded organizational memory.

The experience factory approach is in its basic form very abstract and conceptional [HOU99]. In order to apply it, it is necessary to define its specific goals, the tasks and processes of the involved agents and to install a (technological) platform.
The experience factory approach has been applied in different applications, here we modified the model in order to apply it in the domain of Knowledge Discovery in Databases (Figure 1).

The approach proposed by Basili has been tailored according to [ANT97]. We also distinguish the project teams, conducting different KDD-projects, and the experience factory organization, according to [HOU99], with the roles of the Experience Engineer, the Experience Factory Manager and EF-supporting-agents. See also [BT98] for a similar differentiation.

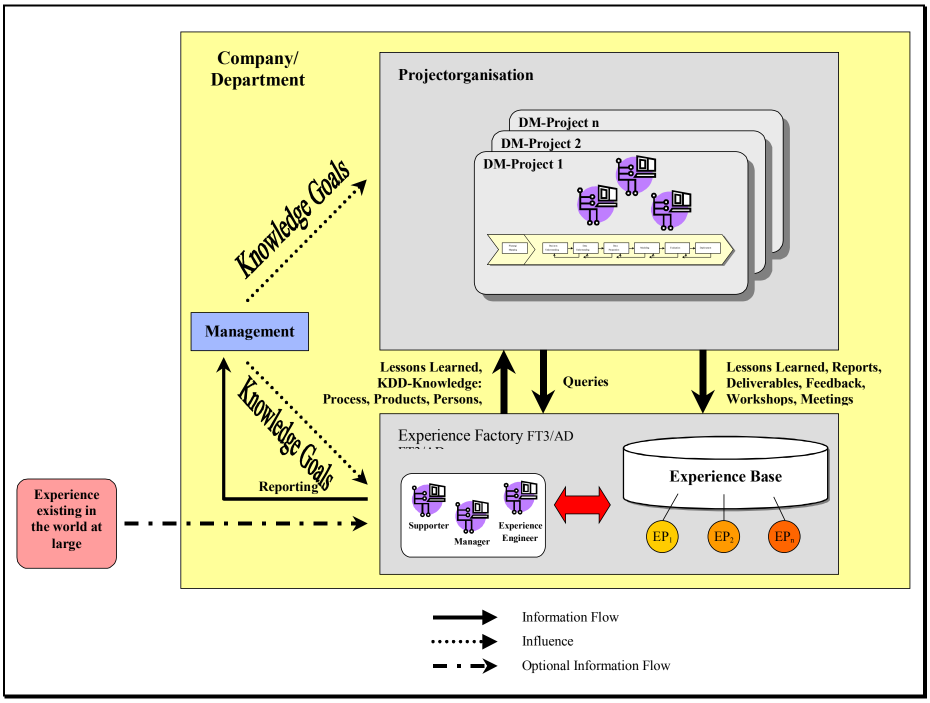


Figure 1: Experience Factory in Knowledge Discovery in Databases.( See also [ANT97])

## 3 The building blocks for Knowledge Management within the Experience Factory

The building blocks according to Probst [PRR99] build a general framework for Knowledge Management and is based on a 2-layered learning cycle. The outside cycle consists of the elements goals, realization and valuation and describes a traditional management control cycle. The inner cycle is represented by the blocks of knowledge identification, knowledge acquisition, knowledge development, knowledge distribution, knowledge use and knowledge preservation. The blocks represent an overall approach for management of knowledge in a business organization. Since the conceptual approach by Probst et al. incorporates the whole organization, the different concrete knowledge managing initiatives have to be fitted in this general approach. Here the Experience Factory should represent a concretization of these blocks. We therefore mapped the approach in order to investigate, how the EF fulfilled these. Further, an approach should be integratable, problem-oriented, understandable, action-oriented and give instruments. Here we want to introduce, how the experience factory and the used CBR-approach realize these requirements.

In the inner cycle, the blocks of knowledge use and knowledge development is in the scope of the project organizations and teams. On the other hand, knowledge identification is one major task of the experience engineers, but depends on the help of the project teams. The EF-supporting agents are responsible for assisting the project-teams and the knowledge distribution. This can happen through joining the project teams, helping through seminars in our so called KDD-Shop or last but not least through our experience base called Core-DM (Case Oriented Reuse of Experiences in Data Mining). Here our department FT3/AD plays an experience factory-like role for the different departments conducting knowledge discovery in databases in corporation with us. We represent a competence center in KDD, helping project partners to conduct KDD. As a research department within DaimlerChrysler, we are further interested in the development and application of new KDD technologies. We participate in KDD research and present the results to leading academic conferences. But on the other hand, we conduct knowledge acquisition through the buy in and evaluation of products, through the cooperation with universities and hiring of new personnel. At FT3/AD we installed the KDD-Shop, where we evaluate new tools and train our own teams and that of the project members in order keep track with the state-of-the-art in our domain.

Our experience engineers are responsible for the documentation of experience packages and artifacts of projects. This is done in cooperation with the project team and according to the EF-management's formulated operative knowledge goals of what types should be collected and how the infrastructure and processes should look. They are therefore all the persons responsible for knowledge preservation. Further, the EF-supportive agents take part in the collection of information and experiences within the project teams.

In the outer cycle, the responsibility for setting knowledge goals can be found on different levels. Probst et al. differentiates between levels for formulating knowledge goals, of interest are more the operative ones. Here realistic goals have to be formulated and further, measures to value these have to be defined and evaluated, closing the loop with the formulation of optimized knowledge goals.

It can be seen, that the basic roles and responsibilities of the Experience Factory can be assigned to the KM building block approach in the context of our department FT3/AD and Knowledge Discovery in Databases (see also figure 2). Although the experience factory approach has its focus on collection and reusing experiences in project work, it also covers with its roles the basic blocks of an general knowledge management approach. While the Experience Factory is more of an organizational approach, giving roles to the different persons, we now want to introduce a more technical approach for completion.

## 4 Cognitive Sciences View: Case Based Reasoning

The approach of case based reasoning (CBR) and knowledge management share the same goal: the use and development of knowledge.
While one can understand under knowledge management a general and large area, incorporating different methods

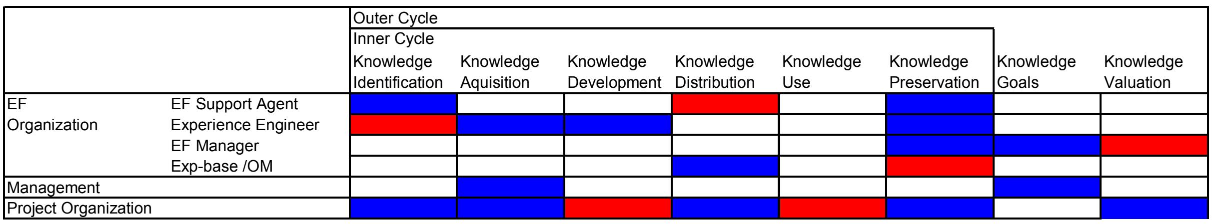


Figure 2: Assigning the responsibilities of the EF organization to the building blocks of Knowledge Management of Probst et al.. (grey = important role, black = less important role).

and techniques, i.e. from organizational and technical, case based reasoning represents a very concrete approach for these mentioned goals.
As we did this in the last section with the experience factory, we will now introduce the case based reasoning approach and show, how it can be used in the general Probst framework and how the building blocks are covered by CBR.

The basic idea of case based reasoning is, that for solving a new problem, a concrete similar but solved solution is tailored to the new context and reused [WES96]. It is

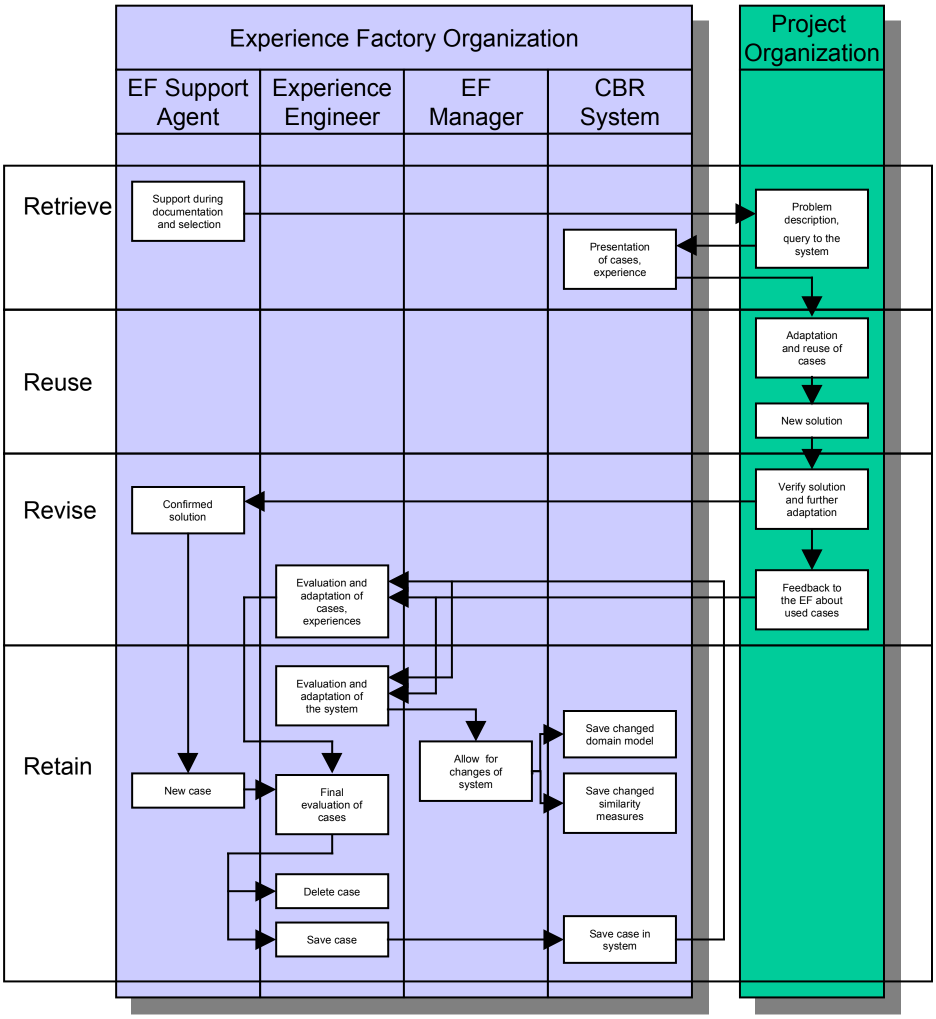


Figure 3: General tasks of the KDD experience factory along the CBR-process.

| CBR-Cycle | | Outer Cycle / Inner Cycle: Knowledge Identification | Knowledge Aquisition | Knowledge Development | Knowledge Distribution | Knowledge Use | Knowledge Preservation | Knowledge Goals | Knowledge Valuation |
|---|---|---|---|---|---|---|---|---|---|
| Run-time | Retrieve | | | | | | | | |
| | Reuse | | | | | | | | |
| | Revise | | | | | | | | |
| | Retain | | | | | | | | |
| | CBR-System Design | | | | | | | | |

Figure 4: Mapping the CBR cycle onto the KM building blocks (grey = many similarities, black = some similarities).

based on a learning-cycle, including its phases retrieval, reuse, revise and retain of cases and experiences [AP94]. It is based on cognition-psychology, stating that experts tend to reuse concrete experiences rather than to solve new problems from the ground up. Case based reasoning tries to realize this idea by describing a problem and its solution by a set of attributes and saves them as a case into a case- or experience base. Besides this knowledge in the experience-base in the form of cases, it is necessary to formulate general knowledge on how to select, interpret and transform cases, i.e. to formulate similarity measures or how to transform the old solution into the new one.

## 5 The Experience Factory as organizational framework for realizing a case based reasoning system

Through the Experience Factory, a case based reasoning system can be given a organizational framework [ABT97]. With this framework, it is possible to compensate the organizational deficits of the CBR approach and assign responsibilities within the CBR-learning cycle:

First of all, the EF-supporting agents, together with the project-teams, are responsible for collecting cases that are candidates for being saved into the experience-base. They give these to the experience engineer for further documentation. On the other side, they are responsible for supporting the project teams by formulating queries to the experience base and for retrieving old cases. They build the interface between the project teams and the EF organization.

the necessary knowledge goals of what is to be reached with this approach and how its success can be measured.

The most important part of a CBR-system is the experience base, where the cases are saved in the form of experience packages. The experience packages are accessed during the retrieve phase using a similarity based measure. In most cases a technological platform exists, in order to do this in an easy and fast way. In figure 3, the lifecycle of a case can be seen along the CBR-phases and the responsibilities according to the experience factory organization.

In figure 4 we mapped the CBR cycle onto the KM building blocks. It can be seen that the CBR cycle by [AP94] corresponds to the realization of the inner knowledge management cycle according to Probst et al. But also the design, evaluation and maintenance of a CBR-system are important topics that need to be covered by an overall approach. Here, we see the EF-manager and the experience engineer responsible for the development of the system, i.e. of the domain model, the structure, the similarity measures and its technical implementation.

In figure 5 we assigned for each of the EF roles the different CBR-phases and added the missing building blocks. This combined framework of experience factory and case based reasoning now covers all necessary steps of a major knowledge management framework, making it to a concretization of the introduced KM-building blocks.

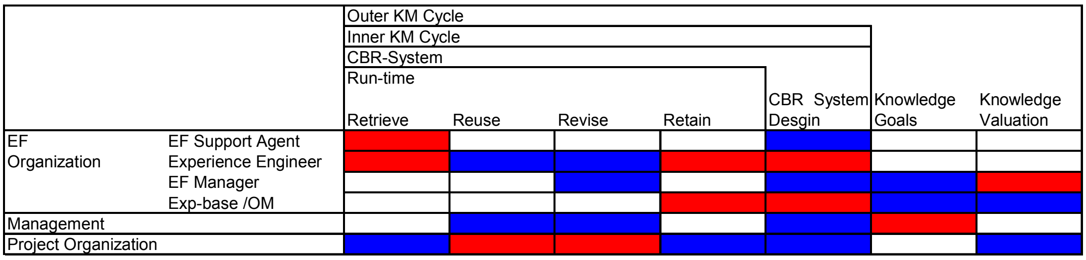


Figure 5: The experience factory roles and case based reasoning for the realization of knowledge management (grey = important role, black = less important role).

The experience engineer is responsible for the final structure and form of the cases. He is a safeguard that the quality of the cases are adequate. Further he has to evaluate and perform maintenance operations on the experience base and its cases. If necessary, he alters the similarity measures for improved retrieval performance or changes the case-structure. On the other hand, the EF-management, together with the rest of the EF-team, sets

## 6 Representation of KDD experience in a case based reasoning system

In a case based reasoning system, knowledge is saved in so called Knowledge Containers, which are case-base, structure/vocabulary, similarity measures and transaction knowledge [RIC98].

The development of a CBR system starts with the structural description of the application domain. This includes the kind of cases one wants to describe, their structure and the definition of their attributes. Further, a similarity measure has to be defined for retrieval from the experience base. As a last step, knowledge on how to transform an old solution to the current situation can be included through transaction rules, but in our case the transaction has to be performed by the user of the system without technological help. The whole structural description of a domain is called domain model and is based on the following primitives [WES96]:

- ***Attribute*** and ***type***s, describe features of a domain (i.e. Text, Reals, Integer)
- ***Concepts, objects***, describe concrete entities of the domain
- ***Relations*** describe the relationship between objects
- ***Rules***, describe rule-based relations between objects

Based on the structural description of the domain, a similarity measure is defined in order to retrieve similar cases from the case base. For each attribute of a given case and a given query, a similarity can then be calculated, which are aggregated to an overall similarity score between a case and a query. The most similar cases can then be presented to the user of the system. Using as similarity measure makes it possible to find not only completely fitting packages, but also "near-matches", which is in the sense of CBR.

We further used keywords to describe the packages textual components. The keyword concept allows the introduction of additional context description and assists the user to identify useful packages. Rather than relying on the experience engineer to find good keywords, we combine our structural CBR approach with a textual CBR technique (tCBR) for the representation of the knowledge of the textual parts (See also [BL00]). Here we rely on the structured form of the cases and use the textual components to extract Information Entities (IEs) about the packages. The knowledge for identifying the IEs of the packages is given by a set of term indices, thesauri, a product/name-index and a term-generalization-index. The content of the dictionaries is collected by our domain experts or automatically by parsing KDD related documents. For retrieving cases, we distinguish the attribute part, where we can make use of the structured domain model's predefined attributes and their possible values, and the textual part, which makes use of domain-dependent and common knowledge stored in the index-vocabulary, thesauri and term-generalizations. For the textual parts, a query to the experience base should give results similar to a package, that contains similar expressions in the form of the IEs. The resulting overall similarity is then calculated as a weighted sum of the similarities of all attributes. Before the experience base can be queried, the packages' IEs have to be pre-calculated. This is done in an off-line process.

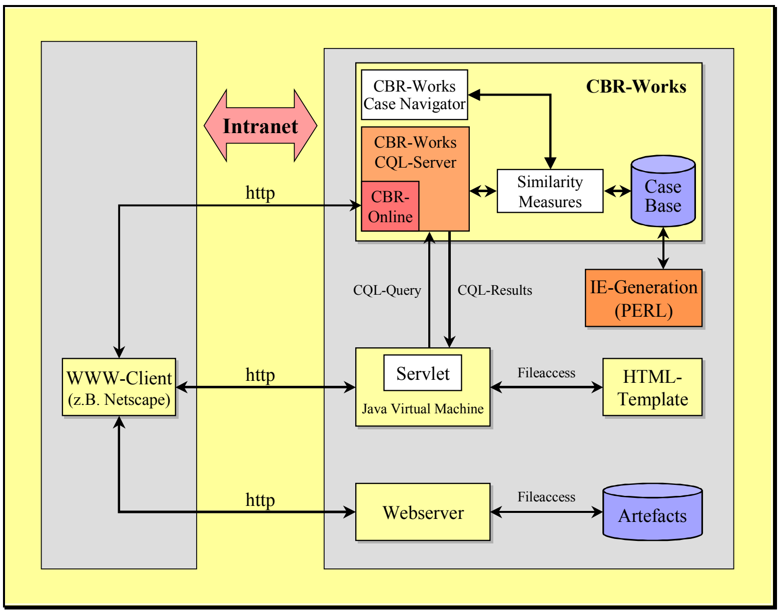


Figure 6: Architecture of Core-DM.

## 7 Technical View: Realization of a CBR based Experience Factory in the case of Knowledge Discovery in Databases

At DaimlerChrysler, KDD is applied from different KDD-teams in projects from Credit-Scoring to Customer Relationship Management. We see KDD as a knowledge-intensive and weak-structured process, where the agents have to choose on each step from a variety of options based on their background-knowledge with KDD. Further we observed, that because of the repetitive application of a standard-process model in KDD, CRISP-DM[1], experience can be used in successive projects. This makes organizational team support an important topic in the case of KDD. Systematic knowledge creation, capture, organization and use provides a way to support the KDD-process model CRISP-DM. We therefore identified types of knowledge that can improve KDD-processes and ways on how experience can be integrated using a CBR-based experience factory for KDD, Core-DM.

On the organizational side, we implemented the proposed CBR based experience factory approach with its processes. The technical architecture of the Core-DM system can be seen in Figure 6 and is based on the commercial tool CBR-Works from TecInno. We implemented an intranet-interface using java-servlet technology, which communicates with the CBR-Works-Server using the CQL-case query language. Further, the EF-teams use the CBR-Works Case-Navigator to author the experience base. Since the user can access different artifacts like KDD-reports, presentations or streams of our Clementine Stream Library[2], we further installed a simple web-server.

We derived nine types of experience packages to be stored in the experience-base and disseminated through the factory (Table 1). We use an object-oriented package-model including generalizations so that common attributes are shared by different package types. In the next section we will introduce three of the nine package types in more detail. These package types represent the different classes of packages used in the experience base, from very structured information packages (i.e. artifacts), to semi-structured packages using large textual component (i.e. lessons learned) in order to represent the knowledge. So far we collected over 350 packages by evaluating different KDD-projects and our KDD-documents like guidelines and handbooks.

### 7.1 Lessons Learned-Packages

Experience Packages of this type describe solutions experienced in a concrete setting of a project (See figure 7). The packages are structured in a part for classification, a main part of a solution-description, and a part giving reasons for this solution (See also [HOU99]). For a first classification of the package, attributes describing a project[3] and the KDD-step, where it occurred, are used. Especially of interest is the step in CRISP-DM, where the package has been used or has been created. This is being modelled by a taxonomy of all possible process phases and steps and indicates in new projects, where they can be reused.

A further context specification is saved additionally to each package. The KDD- and application-context description attributes help to characterize the context of the packages. These features include information about the overall goal of the KDD-Project (i.e. Prediction or Description of data), the KDD-problemtype (i.e. Regression, classification or segmentation of data), and information about the application context. In this case, we applied KDD in the area of marketing and credit-risk-management and specify the concrete application within a taxonomy of these areas. This context also includes features about the objects being described by the data (i.e. private customer information or small commercial

| Experience Package Type | Contains Experience about |
|---|---|
| Documents | Documentation, code, reports required by CRISP-DM |
| Process | Process-model steps definitions used in a project |
| Data | Attributes and data-transformation used in former KDD-projects |
| Product | Product-description of KDD-tools |
| Solutions, Lessons Learned: KDD, Management | Problem/solution pairs, success-factors, mistakes, best practices |
| Experts | Persons involved in KDD projects and skill-database |
| Methods, Techniques | KDD-methods and technique description i.e. neural nets |
| Project | Project-characterization, KDD-problem type, goals, persons involved |
| Formula | Error measures, quality measures |

Table 1: The experience package types used in the KDD-experience factory

[1] Cross Industry Process for Data Mining, see www.crisp-dm.org

[2] Clementine is a KDD Tool by SPSS Inc. used by FT3/AD. Clementine programs are called streams.

[3] Projects are described by its own package type not further described here.

| Domain model part | Description | Attributes |
|---|---|---|
| Context<br>Applied Methods<br>Involved Object | This part of the package describes the context, in which a problem occurred. This includes the selection of predefined dimensions, i.e. KDD-processes task, the KDD-problem type, the used tools, the level of specialization, the methods applied, the specific domain etc. Further, keywords from the keyword-list help to identify the concrete problem. Here tools, methods and objects are described which were applied or involved during the step. They are mostly predefined through the domain model. | Application(Taxonomy of Domains)<br>Data Mining Problem Type(Set)<br>Keywords (String)<br>Objects involved in Experience (Set)<br>Problem class(Set)<br>Project characerization( Subconcept with context-attributes,: Team size, Duration, Region, Tools used, Data sets used)<br>KDD-Goals(Set)<br>Data Mining step in CRISP-DM (Taxonomy)<br>Lessons learned type(Set) |
| Abstract | Here tools, methods and objects are described which were applied or involved during the step. They are mostly predefined through the domain model. | String |
| Problem/Topic | In this section of an experience package describes the problem/topic that had to be solved during the execution of a KDD-step. | String |
| Solution | Here a case/solution or experience description is presented, that can give help in the given context. Further, a justification or rationale, why it has been chosen, can be described. | String |
| Rationale | If it is possible, it describes reasons that made it necessary to perform this step. | String |
| Outcome | In this section, the outcome and result of applying the solution to the problem is described. Further, it is assessed, if the solution is a success for this problem. Note, that also negative outcomes add to the knowledge about a problem. | String |
| References | Since experience packages are only compact documents, links to other information sources or persons can be given. | String |
| Admin | Here administrative information for experience controlling is being given, i.e. number of accesses and ratings. | Author (Reference to Person Experience Package),<br>Comment (String)<br>Controlling Concept (3 Attributes)<br>Knowledge view concept (Review form(Set)<br>Specialization of Experience(Set)<br>Lifecycle of experience) |

Figure 7: Structure of the Lessons Learned packages.

customers) and the regional setting of the application.
The content of the packages is further specified by two attributes. A set of involved objects (i.e. Person, Time, Data or Product) and the class of problem (taken from the areas of management, technical problems, KDD-related problems) help to differentiate the cases. On a knowledge-perspective, three features represent the origin of the package in respect to KDD and its processes (General about KDD, from a KDD project, review of a project), the specialization of the experience (General, special and cookbook), the lifecycle (theory, observation and practice) and the view onto the experience( i.e. Application Developer, Business Analyst, KDD Engineer or End User).
An important characterization is the type of the case, here we distinguish experience between *Best Practice, How To, Mistake/Critique* and *Success Factors.*

The experience is described in the main part of the packages. This is being done in the two text fields, named *topic/problem* and *solution*. So the case information has to be processed in order to fit into these fields. Further, if it is possible, the rationale for applying the solution and the outcome after application can be collected in two further text-fields. The introduced information entities (IEs) are calculated over these four fields in order to use textual CBR techniques. Figure 8 and 9 show, how the experience base can be queried within our departments intranet.

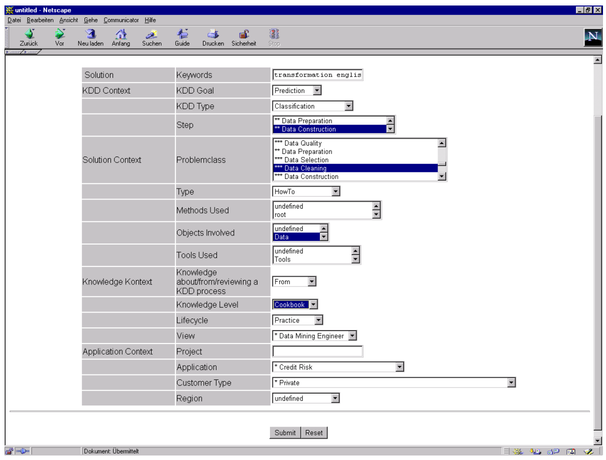

Figure 8: Query for the Lessons Learned Packages of Core-DM. The structural CBR approach allows for the specification of attribute values, the textual approach allows for keyword-search of the packages textual components.

### 7.2 Artifact-Packages

In these packages artifacts of different KDD-processes and projects are collected for reuse. These artifacts can be of different types, i.e. presentations, reports of projects or code-fragments. The user of the system can, therefore, specify the type of artifact he wants to retrieve. This information is represented by an artifact-type in the experience base. Further, the KDD- and application-context specification is saved as before in addition to each artifact-package, and last but not least the CRISP-DM process step.

The artifacts are further described by a short abstract, a detailed description and the project it has been created in. The content of the artifact is characterized by an attribute using a taxonomy of content-types. Here one distinguishes broadly between the result of the KDD-projects like deliverables, reports, process supporting documentation (user-guides or reference-models) for a certain application area. Last but not least, a reference to the concrete artifact is used, so that it can be downloaded from our web-server.

### 7.3 Person-Packages

With this concept the information and especially skills of persons involved in our KDD-projects can be described. The description can be separated into two parts. First, information about the person is being saved as in any person-register, from names to addresses and phone numbers.

In the second part, the skills and roles of a person are described, making it possible to find persons according to their knowledge and expertise and who are willing to share these with others. Rather than using free-text fields to describe these, the packages domain model gives predefined attributes in order to characterize the person. Here the package distinguishes between the KDD-application (i.e. credit scoring for new customers) a person is involved in, its regional setting, the KDD-methods and techniques (i.e. regression techniques) he is expert in. Of further interest in the context of KDD are programming language or product skills, given by a fixed taxonomy.

In order to substitute our departments personal register we also collect traditional individual and person information in free text fields.

## 8 Conclusion

In this paper we introduced our approach for managing experiences in KDD-projects. It is based on the experience factory organization and the approach of case based reasoning. We therefore investigated how the CBR based experience factory approach covers the different aspects of knowledge management, represented by the approach of Probst et. al. It showed that case based reasoning and the experience factory approach complement each other on the technical and organizational level for our needs. We then introduced our realization of the approach in the domain of knowledge discovery in databases. We described our solution for the experience base, called Core-DM, which is based on a combination of structural and textual CBR techniques.

In the next steps we plan to evaluate the system Core-DM. We will derive quantitative and qualitative measures in order to value aspects like quality of the experience-base, economic utility, usability and technical performance. These can then be aggregated to measure the overall success of our knowledge management initiative. A further topic of interest is tightly coupled to this evaluation step. So the maintenance-step of the experience-base and its packages has to be investigated.

## References


[ABT97] Althoff, K.-D./ Birk, A./ Tautz, C.: The Experience Factory Approach: Realizing Learning from Experience in Software Development Organizations. Proceedings of the Tenth German Workshop on Machine Learning, University of Karlsruhe, 6-8- August, 1997. IESE-Report No. 013.97/E. 1997. http://www.iese.fhg.de/pdf_files/iese-013_97.pdf

[ANT98] Althoff, K.-D./ Nick, M./ Tautz, C.: CBR-PEB: Implementing Reuse Concepts of the Experience Factory for the Transfer of CBR System Know-How. Preceeding of the 7th Workshop on Case-Based Reasoning. IESE-Report No. 058.98/E. 1998. http://www.iese.fhg.de/pdf_files/iese-058_98.pdf

[AP94] Aamodt, A/ Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI-Communications, 7(1), 39-59, 1994

[BT98] Birk, A./ Tautz, C.:Knowledge Management of Software Engineering Lessons Learned. Technical Report. IESE-Report No. 002.98/E. 1998. http://www.iese.fhg.de/pdf_files/iese-002_98.pdf

[BCR94] Basili, V.R./ Caldiera, G./ Rombach, H.D.: Experience Factory. In: J. Marciniak, editor, Encyclopedia of Software Engineering, vol. 1, John Wiley and Sons, 1994.

[BAR99] Bartlmae, K.: A CBR based Experience Factory for Data Mining, in: Proceedings of the International Computer Science Conference: Internet Applications (ICSC'99), Lecture Notes in Computer Science, Springer-Verlag, New York, 1999

[BL00] Bartlmae, K./ Lanquillon, C : A KDD Experience Factory: Using Textual CBR for Reusing Lessons Learned, in: Proceedings of the DEXA2000

(Database and Expertsystems Application Conference), London, Lecture Notes in Computer Science, Springer-Verlag, New York, 2000

[HEI98b] Heisig, P.: Projektmanagement und Wissensmanagement. Wissenstransfer noch kein Thema. In: IT Management 7/1998. In german.

[HOU99] Houdek, F.: Empirisch basierte Qualitätsverbesserung. Systematischer Einsatz externer Experimente im Software Engineering. Dissertation. Logos-Verlag. Berlin. 1999. In german.

[KPMG98] KMPG Management Consulting, Parlby, D.: Knowledge Management. Research Report 1998. http://www.kpmg.co.uk/kpmg/uk/services/manage/research/knowmgmt/knowmgmt.pdf

[PRR99] Probst, G./ Raub, S./ Romhardt, K. Wissen Managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Gabler. Wiesbaden. 1999. In german.

[REI98] Reinartz, T. et. al.: The Current CRISP-DM Process Model for Data Mining. In: Maschinelles Lernen. S. 1-9. 1998. In german.

[RIC98] Richter, M.: Introduction (to CBR) In: Lenz, M./ Bartsch-Spörl, B./ Burkhard, H-D./ Wess, S.: Case Based Reasoning Technology. From Foundations to Applications. Lecture Notes in Artificial Intelligence. Springer Verlag, New York. 1998.

[WES96] Wess, S: Fallbasiertes Problemlösen in wissensbasierten Systemen zur Entscheidungsunterstützung und Diagnostik: Grundlagen, Systeme und Anwendungen. Dissertation. Dissertationen zur künstlichen Intelligenz. Bd. 126. Infix Verlag. Sankt Augustin. 1995. In german.