Corporate Information Network (CoIN): Experience Management at IESE

Dipl.-Inform. Björn Decker, PD Dr. rer. nat. Klaus-Dieter Althoff, Dipl.-Inform. Markus Nick, Dipl.-Inform. Andreas Jedlitschka, Dipl.-Inform. Jörg Rech

Fraunhofer Institut für Experimentelles Software Engineering (FhI IESE)

{decker|althoff|nick|jedl|rech}iese.fhg.de

Sauerwiesen 6, 67661 Kaiserslautern

Corporate Information Network (CoIN): Experience Management at IESE

Abstract: *Experience Management (EM) is an area that is increasingly gaining importance. Its roots lie in Experimental Software Engineering, in Artificial Intelligence, and in Knowledge Management. In this paper, we will present the Corporate Information Network (CoIN) initiative, which is responsible for the build-up and operation of Experience Management at IESE. Three aspects of CoIN are detailed: (1) representation of business process descriptions, (2) representation of Lessons Learned, and (3) maintenance of CoIN.*

1 Introduction

In all emerging areas of business and engineering science, there is normally a lack of explicit knowledge about their underlying processes, products, and technologies. Usually, such knowledge is based upon implicit knowledge gained through individual learning from the experience of the people involved. However, this knowledge gained individually may not be accessible to the organization, since (a) its existence remains unknown or (b) members of the organization who possess the needed knowledge leave the organization.

The area of organizational learning [Klu99], as one part of knowledge management, tries to increase the effectiveness of individual human learning for the whole organization. Besides improving internal communication (group learning), organizational learning also includes documenting relevant knowledge and storing it (for later reuse) [AB+98, vH+96]. A learning organization's objective is to enable its members to effectively deal with situational requirements, taking into account past experience.

An approach known from software engineering called Experience Factory (EF) [BCR94] goes one step further. Knowledge (in the form of processes, products, and technologies) is enriched by explicitly documented, real-world experience (e.g., lessons that were learned during the practical application of the knowledge). The EF approach includes capturing, documenting, storing, and disseminating such experience. The EF approach tries to explicitly rebuild human "learning from experience" to further support organizational learning.

We call the research area dealing with both the organizational and the technical support for "learning from experience" Experience Management (EM), which is in accordance with similar suggestions from literature [Tau00, Ber01].

This paper presents the Corporate Information Network (CoIN) Initiative, which has the objective of installing and operating an Intranet-based Experience Factory and thus, Intranet-enabled EM at Fraunhofer IESE (Institute for Experimental Software Engineering). This initiative was started in summer 1999 and has been in continuous operation since summer 2000. This paper presents CoIN as follows: In the subsequent section, the Experience Factory as the theoretical background of CoIN is introduced. Second, the overall picture of CoIN is described followed by two sections detailing the services of CoIN. The next section presents how CoIN is maintained. A section that summarizes this paper and gives an outlook to upcoming work closes the paper.

2 Implementing EM: The Experience Factory

The basic idea of Experience Management is that knowledge within the organization has to prove its value in the business of the organization. In particular, knowledge either (a) has its origin in the business of the organization (i.e., Lessons Learned) or (b) is external knowledge infused by evaluating and adopting it to the needs of the organization. Organizations operating in areas with a high amount of innovation (i.e., eCommerce) will focus more on (a), since there is a lack of widely accepted knowledge. Organizations operating in rather settled areas will focus on (b).

This basic idea includes a continuous stream of experience from the organization's business being recorded and integrated into the existing experience. To do so, with each experience, context (where this experience has been observed before) and validity (how often this experience has been observed) is explicitly stored.

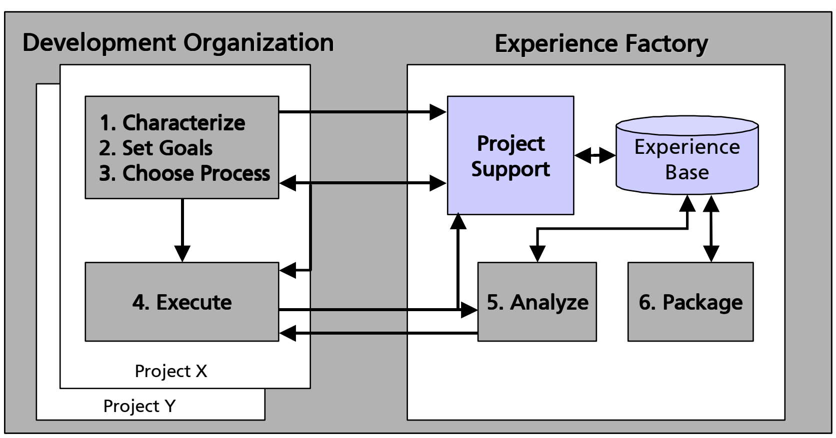

Figure 1 Overview of the Experience Factory (numbers denominate the steps in the QIP)

One generic approch to implement Experience Management is the Quality Improvement Paradigm [BCR94]. Its six steps are depicted in Figure 1: During Project Planning, the current context is *characterized*, the *goals* of the project are *set*, and the *process* to achieve those goals in the current context is *chosen*. Both goals and process can be based on experience from past projects. Their validity for the current context dertermines the risk of application in the upcoming project. During *execution*, project results are analyzed to give input to project management and to record experience. Furthermore, available experience is offered to the projects. After the project has been finished, all results are *analyzed* in a retrospective. Subsequently, the project results are *packaged*: the validation of existing experience and new experience is processed for future use.

To ensure that the experience management activities are performed without negatively affecting the project teams, a separate organizational unit is established: The Experience Factory (EF, see also Figure 1). It takes care of the analysis and packaging of project results, since these steps are not part of the project work. Furthermore, EF supports the projects by offering experience and supporting the adoption of those experiences. The experience itself is stored (virtually) within a single point "inside" the EF: the Experience Base (EB).

By explicitly regarding context and validity, the EF is able to address two common organizational problems: Myths and information overflow. First, the risk of building *myths* (as unjustified acceptance of existing knowledge) within an organization is lowered, since each experience has to undergo continuous evaluation and the applicability of knowledge to a current situation can be determined objectively. Second, *information overflow* can be addressed using (a) the context to identify experience appropriate to the current situation and (a) using the validity of matching experience to evaluate the quality of experience. Therefore, the better the context is defined for the search as well for the stored experience, the better matching experience can be derived.

3 The Services Of CoIN

As mentioned in the introduction, the objective of CoIN is to install and operate an EF at Fraunhofer IESE. In this section, CoIN is described in a top-level view and its current implementation status is presented.

Figure 2 presents the Experience Management Content Framework [JAD+01]. It is the underlying concept of the relationship of services of CoIN and their content. The framework consists of four components: Presentation Layer, Repository, Communities of Practices and Maintenance Component. The *Presentation Layer* is the interface of CoIN to the regular user. It provides (a) uniform access to the information residing within CoIN, (b) stores the user preferences and settings, and (c) aggregates information within CoIN based on those preferences. The *Repository* contains the explicitly captured and consolidated experience of an organization. Currently, a combination of business process descriptions and Lessons Learned was chosen as a starting point, since this combination is likely to reveal synergies [DA+01]. Furthermore, business processes and Lessons Learned could act as a starting point for further EM activities [DJ01]. The *Communities of Practice (CoP)* component is a forum for the members of an organization to discuss current problems, questions and open issues, and to evaluate the content of CoIN. Finally, the *Maintenance Component* supports maintaining and developing the content of CoIN (i.e., the data within the repository) and the services offered to the organization (via the Presentation Layer).

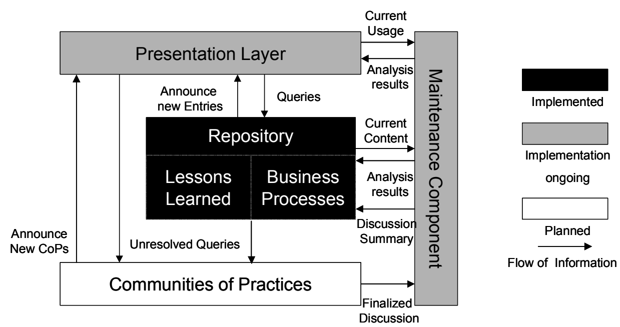

Figure 2 Experience Management Content Framework (EMCF) in CoIN

As also depicted in Figure 2, the Repository, as the core of CoIN, is already implemented and subject to maintenance. Presentation Layer and Maintenance Component are currently under development. The implementation of the Communities of Practice component will be initiated in 2002.

How those components interact with each other is illustrated in a scenario: A Project Manager has to write an offer for a project with customer X. The Presentation Layer provides the Project Manager with the relevant information, in particular (a) with the needed business process descriptions and potential contact persons within the organization and (b) with the Lessons Learned about that customer or similar projects. Furthermore, from other (yet to be implemented) parts of the Repository, (a) offers from similar project conducted in the past are retrieved. If the Project Manager has further questions (e.g., about a certain business process description), the question is transferred to the Communities of Practice. Members of the organization (e.g., process experts) are informed about this question and invited to discuss it. When the discussion is finished, the Maintenance Component analyzes the discussion for experience worth preserving, which is then stored in the repository (e.g., the business process description is reworked).

In the following two sections, the content of the repository is presented in more detail: How business process descriptions and Lessons Learned are represented within CoIN.

4 CoIN-IQ, the Business Process Repository

High quality procedural knowledge, i.e., "good" business processes, is a competitive edge of an organization, since it coordinates its members in an effective and efficient way. Therefore, one of the first efforts within CoIN was to build up a repository of business process descriptions of IESE. As already mentioned in the introduction, this part is called IESE Quality Management System, or IQ for short. The topics currently covered range from core processes (e.g., project set-up and execution) to support processes (e.g., using the IESE information research service) to research focused processes (e.g., performing Ph.D. work at IESE). Based on the results of a diploma thesis performed in advance, the build-up of IQ in 2000 took about ten person months of effort. This effort was equally distributed between the technical realization and the creation of content for IQ. About four person months of effort are allocated in 2001 for further content build-up and maintenance.

This section will present IQ as follows: First, the objectives of IQ are presented. Second, the content of IQ is presented in an overview and in detail, together with the way changes to this content are communicated. Finally, a description of the technology of IQ closes this section.

The objectives of IQ can be positioned according to four criteria: (1) the purpose of process models, (2) the origin and (3) usage of the process models, and (4) the modeling techniques. In summary, IQ uses structured text describing empirical and theoretical process models to be executed by human agents. This is detailed in the following.

For the general purpose of process models, [CKO92] identifies five different categories: Facilitate human understanding and communication, support process improvement, support process management, automate process guidance, and automate execution. According to this classification scheme, IQ fits into the first category of facilitating human understanding and communication: The processes are executed by human agents (i.e., IESE members), based on the process description. Supporting and enforcing process execution beyond this human-based approach (e.g., by workflow modeling and enactment as in [MH99]) was regarded as non-suitable for the purposes of IESE due to the creative nature of its Business Processes. Furthermore, processes according to the process models are executed rather infrequently (< 10 times per month), therefore (a) automation of the processes was not supposed to leverage a high cost/benefit and (b) tracking of process status can be done by asking the responsible process executor. In addition, the experience made with the Electronic Process Guide (EPG) [BV99] showed that web-based process descriptions are a feasible way of distributing process knowledge within creative environments such as Software Business. In particular, changes to web-based process models can be communicated much quicker than paper-based process models, thus enabling quick integration of experience.

The origin of process models can be empirical (i.e., based on actual executed processes) and theoretical (i.e., reflecting a planned process execution). Process models in IQ have boths origins: Some of the process models reflect well-established processes (like, e.g., the administrative project set-up), others represent new procedures (e.g., the reflection of recent changes in the organizational structure of IESE).

The usage of process models can be descriptive (i.e., a description of a process) or prescriptive (i.e., intended to be used as an instruction for process execution). The process models within IQ are prescriptive with different degrees of obligation. In general, administrative procedures (e.g., project accounting) have to be followed without exception; best-practice process models like project management procedures are to be seen as recommendations.

The process modeling technique of IQ is structured text, which is due to several reasons: zero effort training, straightforward modeling and perpetuation in industrial strength applications. Zero effort has to be spent on training, since any IESE member can read structured text without previous training. Furthermore, straightforward modeling means that any IESE members can model processes using structured text, if supported by guidelines and the CoIN team. This aspect is additionally fortified by the experience in scientific publishing of most IESE members. For the modeling itself, any word processing software can be used. Finally, structured text has proven its industrial strength in process models of quality management systems [Dil95].

During the set-up phase, several other modeling techniques were evaluated: (1) extended programming languages like APPLA/A[Ost87], (2) product-flow-based approaches like MVPL [BL+95]), (3) petri-net-based approaches like EPKs [Sch97], and Object-oriented approaches [WMFS00]. All those approaches are well suited for a detailed analysis to gain (a) executable models (e.g., for simulation or workflow execution) and (b) for handling comprehensive, extensive and complex models with many parties involved. However, both (a) and (b) are not applicable to the processes currently captured within IQ, since (a) due to the creative nature of the processes, detailed modeling would have consumed extensive effort, and (b) the number and complexity of process models on the level currently described are rather low. Furthermore, effort would have had to be spent on educating IESE members in one of those modeling techniques or in processing the modeled information into structured text. However, compared to those formal methods, structured text cannot be checked automatically for consistency. Therefore, all process models are reviewed thoroughly by peers and the CoIN team. To facilitate this task, one potential development of IQ could be to enhance the process descriptions by integrating one of those formal modeling techniques that would allow sophisticated consistency checks.

4.1. Content of IQ

To achieve those objectives, the following information is captured within IQ. Each of those information objects can be linked to other objects according to Figure 3:

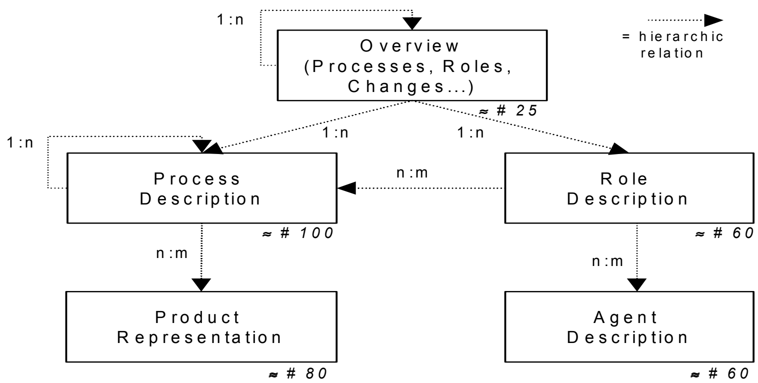

Figure 3 Simplified structure of objects in IQ. Arrows show how objects are linked. The relations are to be read according to the direction of the arrows (e.g., one overview can refer to n other overviews, role descriptions or process descriptions). Italics denominate the number of elements of the respective type of objects within IQ.

- *Process Descriptions*: Process descriptions describe the activities captured within CoIN (e.g., project management). Complex processes are structured into a hierarchy of super- and sub-processes.
- *Role Descriptions*: Role descriptions describe the roles that are involved in the execution of processes.
- *Agent Descriptions*: Agent Descriptions are used within role descriptions to name roles that are performed by a specific IESE member.
- *Product Representations*: A Product Representation represents a document to be used during process execution.
- Overviews: Overviews structure the other objects within IQ to facilitate browsing.

The content of those objects is presented in detail in the following sections, followed by the Lessons Learned of the build-up and operation of IQ.

4.1.1. Process Descriptions

As depicted in Figure 4, a process within IQ is described according to the following structure: "Applicability Information", “Overview of Templates and Additional Information", "Objectives, Results, and Quality Measures", "Actions and Subprocesses" and "Guidelines". The content and purpose of these sections are described in the following:

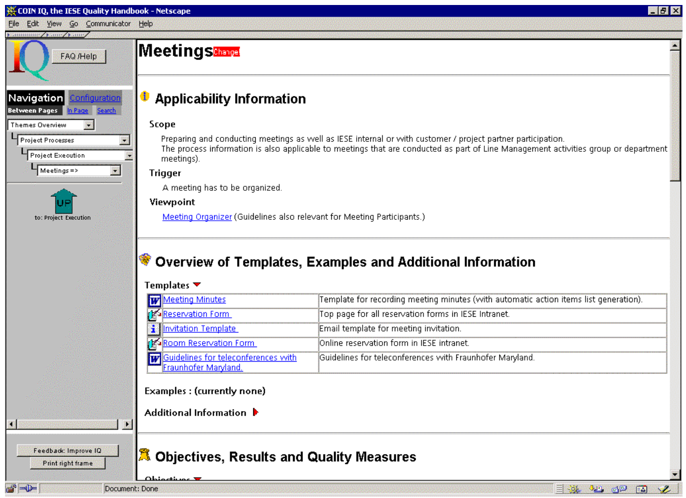

Figure 4 Screenshot of a Process Description. The left frame provides functions applicable to all pages within IQ (help, printing, giving feedback) and navigation functions, including the current position within IQ.

"Applicability Information" gives a short overview of a process' context, thus helping the user to determine if the current process description is the desired one. To facilitate this overview even more, it is again structured into three sub-sections: Scope, Trigger and Viewpoint. "Scope" contains one or two sentences about the thematic range of a process and thus, the content of a process description. "Trigger" as the second sub-section describes the condition that starts the execution of a process. These triggering conditions can be events released from outside IESE (e.g., a customer telephone call), dependencies with other process executions (e.g., start or finish of a process) or dependencies from product states (e.g., a deliverable is about to be finished). "Viewpoint" contains the role from whose view the process is described.

"Overview of Templates, Examples and Additional Information" lists the products referenced by the process description. This overview is intended to support IESE members who are accustomed to the process and just need quick access to artifacts.

"Objectives, Results and Quality Measures" is information intended to guide the execution of a process. The difference between the three sub-sections is the increasing degree of quantification of quality information. "Objectives" are general objectives of the process (see Figure 5 for an example). "Results" are tangible outcomes of the process (e.g., meeting minutes). "Quality Measures" describe properties of such results (e.g., the number of pages of the meeting minutes should range between 10 and 20) or the process itself (e.g., the effort spent on preparing a meeting should not exceed one person day).

"Actions and Subprocesses" describe the steps of the process execution. In IQ, a distinction is made between actions and sub-processes. Actions are atomic steps that are not refined any further. Sub-processes are described in a separate process description according to this structure. The super-process contains a link to the sub-process, followed by a short explanation of the sub-process content.

"Guidelines" give hints for performing a process, like "do's and don'ts" or frequently asked questions about a process. Furthermore, frequently used variances of a process are modeled as guidelines. This reduces the number of similar process descriptions and lowers the effort to maintain the process description. Each guideline has a "speaking headline" in the form of a question or statement, followed by explanatory text.

Currently, those Process Descriptions have a particular function for the integration of services in the EMCF, in particular, knowledge management activities: (a) They are used to define the point in the business processes execution where services (like knowledge acquisition or access to sources) is performed. (b) They describe the usage of services and knowledge management activities themselves (e.g., like project touch down analyses). When the CoP base is implemented, the collaboration will be guided by business process models within IQ. Vice versa, the CoP Base will be used to refine and evaluate the business process models.

4.1.2. Role Descriptions

Role Descriptions in IQ are structured into three main sections. "Role Description", "Processes Referring the Role X", and "Other Roles of Interest". The first section *"Role Description"* contains a general description of the role, who is able to perform this role, who can substitute this role if the respective agent is not available, and which roles support this role in their work. If an agent (i.e., a certain IESE member) can be assigned to one of the latter three sub-sections, the name, telephone and email are stated. For example, the role secretary is performed by several IESE members, each mentioned in this section. The second section *"Processes Referring the Role X"* lists the processes in which the role is involved. Therefore, it serves as a role-specific process overview that allows the performer of a role to gain quick access to relevant processes. The function of the third section *"Other Roles of Interest"* is straightforward; it contains links to other roles that can be relevant to the performer of the current role.

4.1.3. Agent Descriptions

Within the section "Role Description", IESE members who perform specific roles are mentioned with their email, telephone number, and address (see Figure 4). Therefore - besides describing roles themselves - Role Descriptions serve as "yellow pages": A user of IQ can deduce from the role description whether the person mentioned in the contact information is the appropriate one and contact him or her directly. In particular, experts in certain areas are mentioned here to allow accessing the tacit knowledge of such an expert. The other way round, finding roles a certain agent performs, is supported by the

"Agent Overview". Since Agent Descriptions are only used as parts of Role Descriptions, they do not have separate pages.

4.1.4. Product Representations

Products within IQ are mostly referenced by processes. Since they are only referenced by other objects in IQ, they also do not have separate pages like Agent Descriptions. The products within IQ are structured according to two dimensions: representation and usage. Representation is relevant for how the product can be accessed: (1) As an html-document available online, (2) as a downloadable file (e.g., MS-Word file), or (3) as a description of where a document available only as hardcopy can be obtained. Usage of a product can be (a) as a template, (b) as an example document or (c) as additional information to a process or role description.

Those products are a particular area of interest for the CoIN team. In the course of the project analysis - primarily intended to capture Lessons Learned – re-usable documents are identified. If a document is of special interest for IQ users, it is processed further into a template to facilitate its application in future projects. Otherwise, it is brought into IQ as an example or as additional information. Furthermore, experiences with currently available documents are also captured for continuous improvement of those documents.

With the implementation of the organization-wide workspace, IQ can take over those Products Representations via the Information Integration Layer. Currently, the already mentioned loose integration of services by IQ is implemented by embedding them into the process descriptions as products (e.g., the reservation service for meeting rooms.)

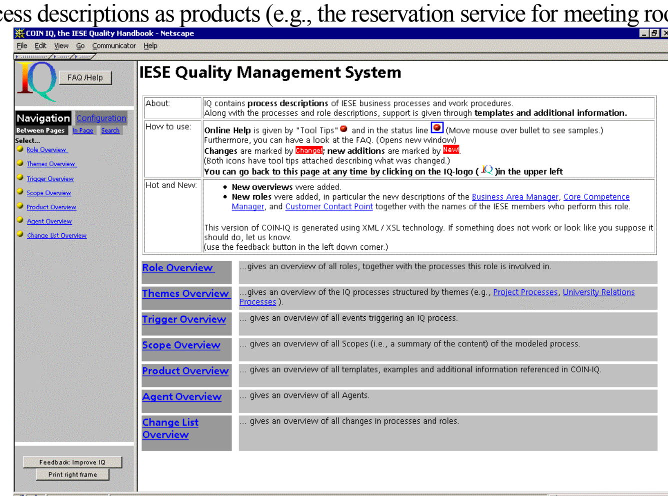

Figure 5 Screenshot of the entry page of IQ as an example of an overview. Further overviews can be reached from this page.

4.1.5. Overview Descriptions

Overviews (see Figure 5) support a user in navigating through the elements mentioned before by (1) structuring the other objects (e.g., overviews for roles or processes of a certain category) or (2) offering a certain view to those objects (e.g., overview of activities throughout the year). To structure the objects within IQ, those overviews are sufficient because (a) the topics of processes and the functions of roles are rather disjunctive and (b) – as pointed out in [Gri94] - the overall number of objects is small enough for structuring it via overviews (see Figure 2).

4.1.6. Change Marking

One particular overview covers (a) the changes in and new additions of objects to IQ. Since each change is mentioned with the date and a short description of the change, a user can gain an overview of changes that have occurred since his or her last visit to IQ. In addition, the most recent changes are mentioned on the entry page of IQ (see Figure 5). This enables quick access to those changes, which lowers the effort for propagating changes within the organization.

Furthermore, changes or new objects in IQ are marked by a "new" or "changed" icon (see Top of Figure 3 and 4). This change marking is sufficient for the information within IQ: Process descriptions are not intended to be read on a daily basis. Therefore, a user will read the whole process description and focus his or her interest on the changed sections.

Besides listing and marking the changes in IQ itself, wide-scoped changes are communicated within IESE via email to all IESE members. When the Presentation Layer is implemented, those changes are aggregated and distributed role-specifically to inform IQ users in a more precise and less intrusive way compared to the current notification via email.

4.2. Technology

IQ is based on XML technology [XML01]: All of the process elements described in the previous sections are kept in an XML file, which is edited using a standard XML Editor. From this file static web pages are created using XSL-style-sheets, which are accessible on the IESE Intranet via an Apache Server. The generation of static web pages is sufficient for the purpose of IQ, since the elements of the whole process model are supposed to change rather infrequently at IESE (maximum once per week, in general one or two minor changes per month).

Furthermore, server-side oo-perl cgi scripts and client-side java scripts are used to provide further browsing and configuration functionality. Since these functionalities affected performance and usability, IQ is currently being re-designed to a pure HTML-based version.

5 CoIN-EF, the Lessons Learned Repository

The second major subject area currently being addressed in CoIN are Lessons Learned (i.e., small grained experience items) about Project Management. This subproject of CoIN is called EF (for Experience Factory) and was the first application developed within CoIN. Lessons Learned can cover different topics and take on different forms [BT98]. Within EF, one Lesson Learned can take on the form of an Observation, a Problem, Guideline, Pragmatic Solution, or an Improvement Suggestion. To store and retrieve those Lessons Learned, case-based reasoning (CBR) technology is used [Tau00].

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 correction is applied after a problem has already occurred.

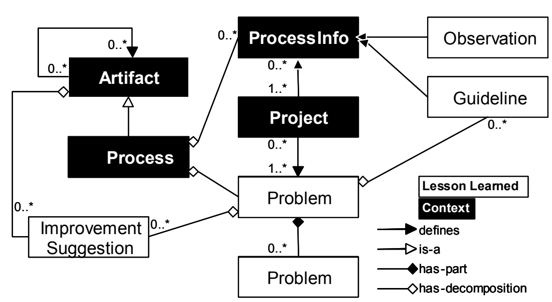

Figure 6 Types of Lessons Learned in CoIN and their context

Furthermore, the context of these Lessons Learned is modeled by the two concepts "Project" and "Process": *“Project”* is a characterization of the project in which the Lesson Learned was gained (e.g., person month, duration). *“Process”* names the business process and thus the project phase in which the Lesson Learned was gained (e.g., Project Set-Up). Figure 6 shows the interrelationships between the context and the different types of Lessons Learned.

5.1. Populating CoIN-EF

The Lessons Learned repository must be filled and updated with new Lessons Learned, in order to build up and maintain the value of the repository [NA00]. Since the elicitation of Lessons Learned cannot be fulfilled automatically due to the complexity of project management, CoIN applies the following steps shown in Figure 4, which is a further development of the steps described in [AB+99]:

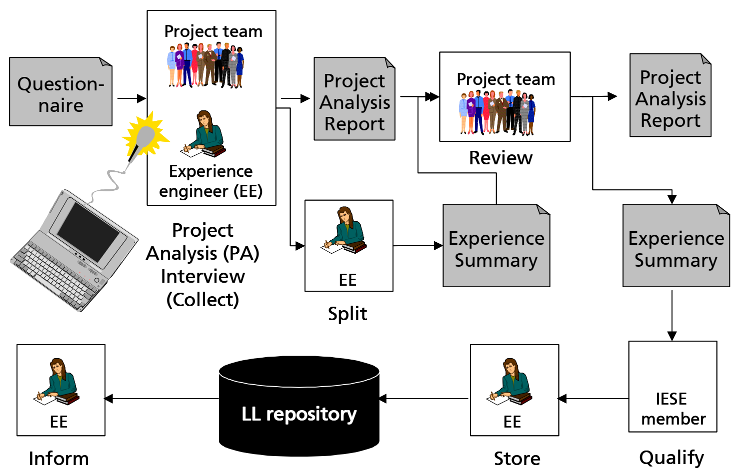

Figure 7 Process for recording Lessons Learned.

- *Collect*: The initial step in experience collection is done by performing two hour project analysis interviews. Those interviews are either conducted at the end of a project [CDF96] or — in case the project has a duration of more than nine months — periodically every six months. The whole interview is recorded using a notebook computer. This recording allows to listen to the contributions of the interviewees on a per-question basis, which facilitates the processing of the interview results into project analysis reports (PARs). A PAR contains an updated characterization of the project, things to watch out for in similar projects, things that went well, and things that the interviewed project team would do differently if it had to do the same project again. Based on the PAR, an overview of the LL to be put into the Repository is created, which is called the Experience Summary.
- *Review*: In order to avoid misinterpretations, and to grant publication permission for the content, the project team reviews the PAR and the Experience Summary. Only the Experience Summary is given to IESE members outside the project and CoIN team.
- *Qualify*: Each Lesson Learned is qualified by analyzing its quality (e.g., its comprehensibility) and checking whether a similar Lesson Learned is already stored in the case base. In addition, a peer review of the Experience Summary is performed to evaluate whether an experience is (a) phrased in an understandable way and (b) worth preserving.

• *Store*: In the next step, the Lessons Learned in the Experience Summary are stored by copying them into the Lessons Learned repository. Since this LL repository utilizes CBR, each Lesson Learned is stored as an individual case: The project characterization in the case base is updated; the Lessons Learned are entered into the case base. If a similar Lesson Learned is already stored, the new Lesson Learned may be rejected, be merged with the already stored Lesson Learned — possibly generalizing its context —, or replace the stored Lesson Learned.
• *Publish*: After the new experience has been stored, it is made available for retrieval, thus enabling the sharing of the new Lesson Learned. Other project teams can now access the new Lessons Learned in the experience base.
• *Inform*: Finally, everybody who may be interested in the new Lesson Learned (i.e., project teams working on a similar project) is informed. This is currently done on a face-to-face basis, in particular during other project analyses.

The Collect and Review step is currently represented in COIN-IQ and referenced from the project execution and project wrap-up process descriptions. The reason for this selection was that the build-up of COIN-IQ concentrated on processes that are of relevance to a large number of IESE members. Collect and Review require the interaction with project teams at IESE, which are the majority of IESE members. The other steps are performed only by the COIN team. However, COIN applies the Project Analysis to itself, thus gaining Lessons Learned that support the description of the other steps in COIN-IQ in the future.

5.2. Technology

To store and retrieve the various types of Lessons Learned, the technical infrastructure of COIN-EF was developed based on INTERESTS (Intelligent retrieval and storage system; [ABT00][AB+99], see Figure 8).

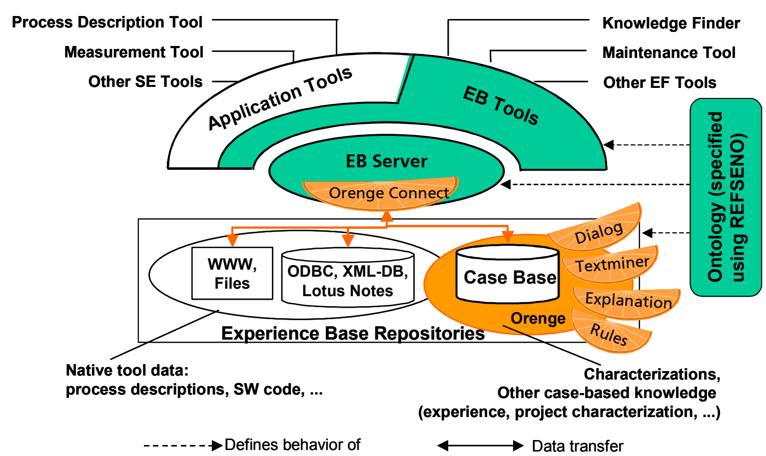

Figure 8 INTERESTS Architecture

This technical infrastructure was systematically developed using the DISER method (Design and implementation of software engineering repositories [Tau00][TA00]). Part of INTERESTS is a case-based reasoning tool allowing similarity-based, context-sensitive retrieval: Orenge (Open Retrieval Engine) from empolis [Emp00]. Case-based reasoning has been recognized as a suitable technology for implementing knowledge management applications (e.g., [GT99][GR99]. This was confirmed for COIN-EF: An experiment showed that using COIN-EF was rated as more efficient than asking colleagues. Furthermore, this experiment showed that most users of COIN-EF would combine querying COIN-EF and asking their colleagues [TA+00].

Besides the Lessons Learned, the context concepts are also modeled as cases. Semantic relationships between cases, for instance the relation of a Lesson Learned to the Business Process part of a context, are represented by references. For example, an observation references the Business Process for which it is relevant and the project in which it was gained (root context). The similarity-based querying facility of INTERESTS/Orenge allows finding Lessons Learned that were captured in contexts similar to a context at hand. Thus, potentially applicable Lessons Learned are identified (even if the application context has not been generalized from the root context yet). All Lessons Learned have the name of their originators attached, which enables a user of COIN-EF to ask the experience provider for more detailed information if necessary.

6 Maintaining CoIN

After CoIN-IQ and CoIN-EF have been described, the maintenance of CoIN is presented in the following. First, this section gives the definition of maintenance and how it is handled within CoIN. Then the maintenance-supporting framework is presented.

Within CoIN, a wide-scoped definition of maintenance is used: The goal of maintenance is to preserve and/or improve the value of CoIN to IESE [NAT01]. The main driving force of maintenance is the CoIN team, a team of six scientists working for CoIN on a part-time basis. The CoIN team either performs the maintenance activities themselves, or distributes them among other IESE members. The latter method will gain even more importance when the CoPs are installed, since the CoIN team will ensure that open questions are directed to the experts within the organization.

Compared to a dedicated, full-time organizational unit performing maintenance, this distribution of maintenance and the part-time basis of the CoIN team demand (a) increased coordination and tracking of the execution of maintenance activities and (b) capture of the knowledge needed during maintenance. The last point (b) also allows to delegate parts of the maintenance activities to lower ranking members of an organization. In the long run, the effects of personnel turnover in the CoIN team are minimized. However, one needs to take the different forms of maintenance knowledge into account:

Quality Knowledge, Maintenance Process/Procedure Knowledge and Maintenance Decision Knowledge:

Quality knowledge describes how the quality of the EB system is measured and the current status of the system with respect to quality as well as the rationale for the definition of quality [Men98]. Quality knowledge deals with quality aspects of the EB/CBR system as a whole, i.e., the EB's contents and conceptual model as well as retrieval mechanisms, usability of the user interface, etc. An example for content-related qualiy knowledge is a definition of measures for the utility or value of single cases [NF00]. There are several types of quality knowledge that are related as follows: The measures define what data is collected. The data collection is performed automatically or manually by respective data collection procedures. The collected data is analyzed using predefined models or procedures. The results of the analyses can be used for justifying an EB and as input for decisions about maintenance [NAT01][NF00].

Maintenance process and procedure knowledge defines how the actual maintenance activities are performed. The actual maintenance can be performed as a mix of automatically and manually performed activities. For the *automatically* performed activities *(maintenance procedure*s), tool support by components of a CBR system or separate tools is required. The remaining activities have to be performed *manually (maintenance processe*s). To improve guidance for the maintainers, descriptions of these processes are provided (e.g., detailed description of the acquisition of new cases through collecting cases, reviewing these cases, and publishing them in the case base, see DISER [Tau00] and INRECA methodology [BBG+99]) for examples). To combine manual and automatic maintenance, a maintenance process can have automated subprocesses/ steps, which use input from or provide input for manually performed steps.

Maintenance Decision Knowledge links the quality knowledge with the maintenance process knowledge. It describes under what circumstances maintenance processes/procedures should be executed or checked for execution. Such maintenance knowledge can be described in an informal manner as *maintenance policies* [LW98], which define when, why, and how maintenance is performed for an EB/CBR system. The "why" addresses not only the reason of maintenance but also the expected benefits of the maintenance operation, which should be related to the objectives of the EB/CBR system or to the general goal of maintenance (i.e., to preserve and improve the EB's value [NAT01]). Since these objectives are typically very high-level, it is not very meaningful to address the EB objectives directly. Instead, we use a refinement of the objectives: the quality criteria from the evaluation program or the recording methods. The "how" is a combination of maintenance processes and procedures with additional steps as "glue."

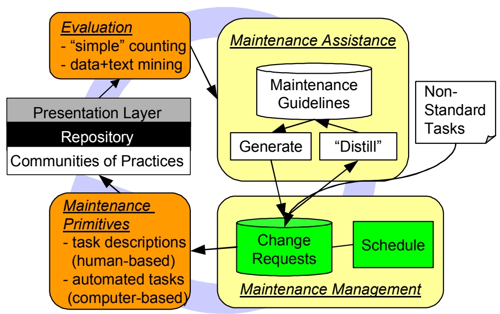

Figure 9 EMSIG Framework

IESE's solution to coordinating experience and capturing the relevant maintenance knowledge is the EMSIG (Evaluation and Maintenance of Software Engineering Repositories) framework as depicted in Figure 9 [NAT01]. This framework includes a method as well as a technical infrastructure and is currently being developed and employed in CoIN. The *evaluation* component supports analysis of the content and usage of services within CoIN, thus, is responsible for the quality and value issues and deals with the "why" of maintenance. The results of these analyses provide the basis and input for making maintenance decisions. The *maintenance assistance* component supports the decision-making task by exploiting the evaluation in order to propose change requests (i.e., basic maintenance activities to be done). This deals mainly with knowledge issues and the "what" of maintenance ("what" to do for "what" knowledge/experience) and has to consider the "why" (justification from evaluation in the form of expected benefits vs. expected maintenance effort). To support the task of learning about maintenance, typical tasks or patterns of maintenance activities are identified and captured ("distill maintenance guidelines"). These maintenance guidelines can be used for generating change requests automatically. The *maintenance management* component supports the task of organizing maintenance and, thus, is responsible for handling the change requests in an appropriate order. When a change request is executed, the *maintenance primitives* component provides the actual methods, technique, and/or tool(s) to perform the basic maintenance activities as demanded by the change request.

7 Summary and Outlook

In this paper, the Corporate Information Network (CoIN) Initiative as the implementation of EM at Fraunhofer IESE was introduced. The Experience Management Content Framework (EMCF) as the top-level view to services of CoIN was depicted and the business process description and Lessons Learned service was presented in detail. Following that, the maintenance of CoIN and its support through the EMSIG-Framework was described.

CoIN will be developed further in the course of several Ph.D. theses currently running at IESE. One significant step towards full implementation of the EMCF will be the BMBF funded project InDiGo [Ind01] This project will develop a technical infrastructure and method for the following aspects: (a) Establishing Communities of Practices about business process, (b) process and project context sensitive presentation of Lessons Learned and (c) application of data-mining technology within Experience Factories [RAD01]. The expected benefits of this project are (a) the support of business process change management and (b) a demand oriented presentation of Lessons Learned. Both technical infrastructure and method will be evaluated in a case study taking place in mid- 2002.

A second application of Experience Management will be the DISER method. By applying the EMCF to this method, Lessons Learned about the application of its underlying processes will be used to maintain and evolve the method.

References

[AB+98] Abecker, A..; Bernadi, A.; Hinkelmann, K..; Kühn, O.; Sintek, M.: Towards a Technology for Organizational Memories, IEEE Intelligent Systems 1998.

[AB+99] Althoff, K.-D.; Birk, A.; Hartkopf, S.; Müller, W.; Nick, M.; Surmann, D. Tautz, C.: Managing Software Engineering Experience for Comprehensive Reuse; Proceedings of the Eleventh Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, June 1999; Knowledge Systems Institute; Skokie, Illinois, USA; 1999.

[ABT00] Althoff, K.-D.; Bomarius, F.; Tautz, C.: Knowledge Management for Building Learning Software Organizations, Information Systems Frontiers; Kluwer Academic Publishers; 2000.

[BCR94] Basili, V.R.; Caldiera, G.; Rombach, D.: Experience Factory; In Marciniak, J.J. ed., Encyclopedia of Software Engineering, vol 1, 469–476; John Wiley & Sons; 1994.

[BT98] Birk, A. and Tautz, C. Knowledge Management of Software Engineering Lessons Learned; Proceedings of the Tenth Conference on Software Engineering and Knowledge Engineering; San Francisco Bay, USA. Skokie, Illinois, USA: Knowledge Systems Institute; 1998.

[CDF96] Collier, B.; DeMarco, T.; Fearey, P.: A defined Process for Project Postmortem Review; IEEE Software 13(4): 65–72; 1996.

[DA+01] Decker, B.;Althoff, K.-D.; Nick, M.; Tautz, C.: Integrating Business Process Descriptions and Lessons Learned with an Experience Factory; Professionelles Wissensmanagement Erfahrungen und Visionen WM 2001; http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-37/Decker.pdf; 2001.

[DJ01] Decker, B.; Jedlitschka, A.: The Integrated Corporate Information Network iCoiN: A Comprehensive, Web-based Experience Factory; Proceedings of the Workshop on Learning Software Organizations; Kaiserslautern; Lecture Notes in Computer Science, Springer, 2001.

[GR99] Göker, M. Roth-Berghofer, T.: Workshop on "The Integration of Case-Based Reasoning in Business Processes"; In: Proceedings of the ICCBR '99 Workshops. Technical Report, LSA-99-03E; Department of Computer Science, University of Kaiserslautern, Germany: Centre for Learning Systems and Applications; 1999.

[GT99] Gresse von Wangenheim, C. Tautz, C.: Workshop on "Practical Case-Based Reasoning Strategies for Building and Maintaining Corporate Memories"; In: Proceedings of the

ICCBR ’99 Workshops. Technical Report, LSA-99-03E, Department of Computer Science, University of Kaiserslautern, Germany: Centre for Learning Systems and Applications; 1999.

[Ind01] InDiGo Homepage; http://www.fraunhofer.ais.de/~leopold/indigo.html, Fraunhofer AiS, Germany; 2001.

[JAD+01] Jedlitschka, A.; Althoff, K.-D., Decker B., Hartkopf, S.; Nick, M.: Corporate Information Network (COIN): The Fraunhofer IESE Experience Factory; Workshop Wissensmanagement und E-Business, GI-Jahrestagung (Informatik 2001, http://www.informatik2001.at/); Austria, Wien; 2001.

[JA01] Jedlitschka, A.; Althoff, K.-D.: Using Case-Based Reasoning for User Modeling in an Experience Management System; to appear in proceedings of the ABIS-Workshop; Dortmund; 2001

[Klu99] Kluge, A.: Erfahrungsmanagement in lernenden Organisationen; Verl. für angewandte Psychologie; Göttingen 1999.

[NAT01] Nick, M.; Althoff, K.-D. & Tautz, C.: Systematic Maintenance for Corporate Experience Repositories. Computational Intelligence 17(2), 364-386; 2001.

[NA00] Nick, M; Althoff, K.-D.: Systematic evaluation and maintenance of experience bases; In Workshop on Flexible Strategies for Maintaining Knowledge Containers at ECAI; 2000.

[RAD01] Rech, J.; Althoff, K.-D.; Decker, Björn. (2001). Using Knowledge Discovery Technology in Experience Management. Submitted to Workshop "Maschinelles Lernen (FGML01)", GI-Workshop -Woche „Lernen – Lehren –Wissen –Adaptivität (LLWA01)“, Universität Dortmund; 8.-12. Oct. 2001.

[TA00] Tautz, C.; Althoff, K.-D.: A Case Study on Engineering Ontologies and Related Processes for Sharing Software Engineering Experience; Proceedings of 12th International Conference on Software Engineering and Knowledge Engineering; 2000.

[TA+00] Tautz, C.; Althoff, K.-D.; Nick, M.; Ochs, M.: An Empirical Investigation on the Effectiveness and Efficiency of Utilizing Repositories for Qualitative Experience; IESE-Report No. 016.00/E, Fraunhofer IESE, Kaiserslautern, Germany; 2000.

[Tau00] Tautz, C.: Customizing Software Engineering Experience Management Systems to Organizational Needs; Ph. D. diss., Dept. of Computer Science, University of Kaiserslautern, Germany; Stuttgart: Fraunhofer IRB Verlag; 2000.

[Tec01] Homepage of tec:inno empolis knowledge management, http://www.tecinno.de/; Tec:Inno, Germany; 2001.

[vH+96] van Heijst, G., van der Speck, R., and Kruizinga, E. Organizing corporate memories. In Proceedings of the Tenth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, SRDG Publications, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4; 1996.

[Ber01] Bergmann, Dr. Ralph: Experience Management – Foundations, Development Methodology, and Internet-based Applications; post-doctoral thesis University of Kaiserslautern, submitted; 2001

[BL+95] Bröckers, A.; Lott, C.M.; Rombach, H. D.: Verlage, M.: MVP-L Language Report Version 2; Technical Report University of Kaiserslautern, Department of Computer Science 265 / 95; 1995.

[BV99] Becker-Kornstaedt, U.; Verlage, M.: The V-Model Guide: Experience with a Web-based Approach for Process Support; Proceedings of Software Technology and Engineering Practice (STEP); 1999.

[CKO92] Curtis, B.; Kellner, M. I., Over, J.: Process Modeling; Communications of the ACM; 1992.

[Dil95] Dilg, P.: Praktisches Qualitätsmanagement in der Informationstechnologie; Carl Hanser Verlag, Wien; 1995.

[MH99] Maurer, F.; Holz, H.: Process-Oriented Knowledge Management For Learning Software Organizations, Proceedings of 12th Knowledge Acquisition For Knowledge-Based Systems Workshop 1999 (KAW99); Banff, Canada; 1999.

[PMI96] Duncan, W. R. (Director of Standards): A Guide to the Project Management Body of Knowledge; Project Management Institute; PMI Publishing Division; North Carolina, USA; http://www.pmi.org/publictn/download/1996welcome.htm; 1996.

[Sch97] Scheer, A.W.: Referenzmodelle für industrielle Geschäftsprozesse (siebte Auflage), Springer Verlag, 1997.

[Ost87] Osterweil, L.: Software Processes Are Software Too, Association for Computer Machinery (ACM); 1987.

[WMFS00] Workflow Management Facility Specification, V1.2, Object Management Group; http://www.omg.org/; 2000.

[XML01] http://www.xml.org; 2001.

[LW98] Leake, D. B. and Wilson, D. C.: Categorizing case-base maintenance: Dimensions and directions. In Smyth, B. and Cunningham, P., editors, Advances in Case-Based Reasoning: Proceedings of the Fourth European Workshop on Case-Based Reasoning, Springer-Verlag, Berlin, Germany. 196–207; 1998.

[Men98] Menzies, T. : Knowledge maintenance: The state of the art., The Knowledge Engineering Review., 1998.

[NF2000] Nick, M.; Feldmann, R.: Guidelines for evaluation and improvement of reuse and experience repository systems through measurement programs; In *FESMA-AEMES 2000*; Madrid, Spain; 2000.

[BBG+99] Bergmann, R., Breen, S., Göker, M., Manago, M., and Wess, S.: *Developing Industrial Case-Based Reasoning Applications – The INRECA Methodolog*y. Springer Verlag; 1999.