Distributed Participative Knowledge Management: The indiGo System

— Methodology — Technology — Evaluation —

Jörg Rech, Björn Decker, Klaus-Dieter Althoff
Fraunhofer Institute for Experimental Software Engineering (IESE)
67661 Kaiserslautern, Germany
{rech, decker, althoff}@iese.fraunhofer.de

Andreas Klotz, Edda Leopold, Angie Voss
Fraunhofer Institute for Autonomous Intelligent Systems (AIS)
Schloss Birlinghoven
53754 Sankt Augustin, Germany
{andreas.klotz, edda.leopold, angie.voss}@ais.fraunhofer.de

ABSTRACT

In globally distributed software organizations, the quality of processes and their models is very important for the quality of the products developed. Nevertheless, many organizations neglect this process knowledge and leave the experience in the heads of their experts. In this chapter, we present the indiGo approach for distributed organizational knowledge management for process learning based on eDiscussions, text mining, experience management, and process evolution methods. The indiGo methodology enables an organization to introduce and continuously evolve their process knowledge in a global setting. It supports an organization in learning about its processes and process modeling techniques, and enables it to record valuable experiences from process users and experts.

Keywords: globally distributed knowledge management, experience elicitation, organizational process learning, process improvement, knowledge acquisition, text mining, eDiscussions

INTRODUCTION

Process models of organizations operating in the software industry are one of their major assets and range from business to software development process models. Especially in the dynamic and innovative software market, they are constantly subject to changes in the business and to scientific advances. To survive these changes, process models need to be constantly inspected, evaluated, revised, and improved by several active process users and process experts. A common understanding and the continuous improvement of a process are of particular importance in spatially and temporally distributed software development organizations. Furthermore, the sharing of process- and technology-related experience needs to be supported by technology, since these groups cannot share their knowledge in conventional ways like meetings or personal talks.

The approach of the indiGo project [1] – funded by the German Ministry for Education and Research (BMBF) – is to increase the applicability of knowledge in the form of process models as well as support their inspection and improvement. It offers members of a spatially distributed organization an instrument to participate in moderated discourses about the structure, content, and execution of process models. One side effect is that the organization will collect lessons learned of their employees while executing the process.

As depicted in Figure 1.1, the process improvement lifecycle in indiGo starts with a plain process model that is annotated, discussed, and enriched with lessons learned, eventually to be revised into the applicable process model based on corporate goals.

One benefit of indiGo is the increased speed of innovation cycles by involving many people and recording more information on processes. It supports an organization and its members in improving and learning about its processes. In addition, indiGo augments the modeling of procedural knowledge through discourses with text mining and case-based reasoning techniques. Current approaches to experience management [2] [6] are reinforced by providing a solution to integrate results of discourses into processes and experience bases.

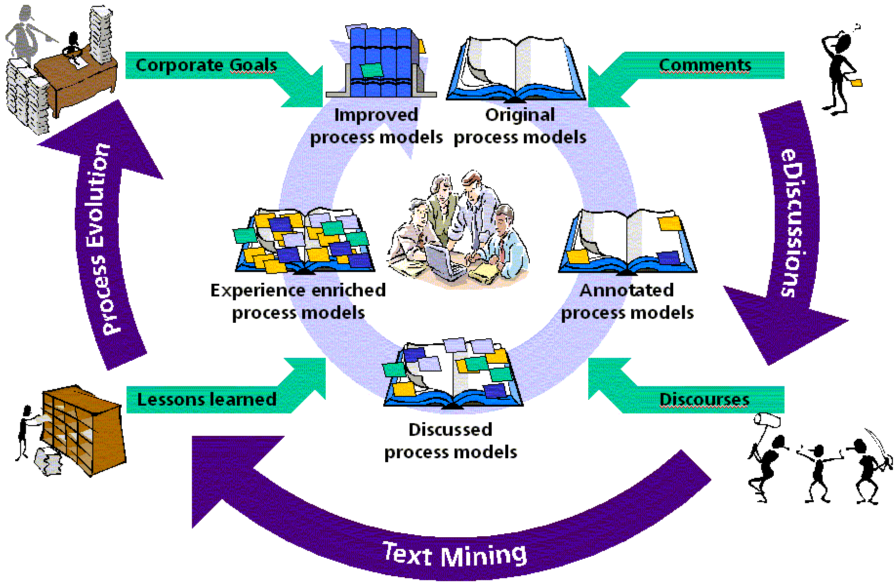

Figure 1.1 Process improvement in indiGo

To support the evolution of process models, indiGo offers an integrated, comprehensive set of methods, and a technical infrastructure. Both the methods and architecture were evaluated in mid 2002 in a case study with various business processes carried out at IESE.

DISTRIBUTED PARTICIPATIVE KNOWLEDGE MANAGEMENT

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 reused carefully, *knowledge* is usually applicable in previously unknown contexts with a fair amount of certainty. Unfortunately, today a few experts who acquired knowledge through their experiences in day-to-day work or research mainly hold the knowledge of an organization.

Participative knowledge management enables an organization to cooperatively record, inspect, and adapt knowledge on a global scale. Elicitation and adaptation of knowledge are the most critical tasks in knowledge management with respect to further reuse of knowledge. The participation of managers, employees, or external experts helps to record different types of knowledge applicable in several explicit contexts, especially in a distributed environment.
In participative knowledge management, four categories of knowledge exist. *Private annotations* help a participant to manage and store private knowledge about a process or product for future reuse. These annotations can be used in *open discussions* in order to discuss or inspect a given product or process. Sharing one's opinions with other users or experts can result in decisions about changes to the product or process, or exposes *experiences* in the form of lessons learned. Finally, experiences about a process or product result in improved and adapted processes or products — knowledge in the form of guidelines, templates, and best practices.

Knowledge compaction, usage and construction

indiGo takes into account all four categories of knowledge and supports them as successive stages in a process of knowledge compaction (aggregation, condensation, summarization or classification). Figure 1.2 arranges the four knowledge categories onto one layer and embeds it into layers of knowledge usage and knowledge construction.

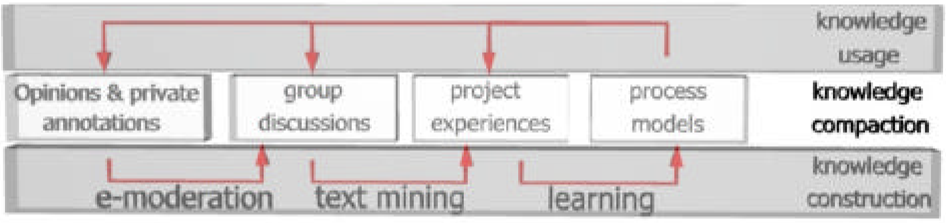

Figure 1.2 Layers of knowledge compaction, usage and creation

Knowledge compaction is the process of (a) decontextualization and (b) formalization with the goal of (c) decreasing modification times as well as increasing (d) lifetime, (e) commitment, and (f) visibility. As indicators of knowledge compaction (a-f) are correlated, they exhibit a clear progression from private annotations via group discussions to lessons learned, and to the organization's process models. Private annotations are highly contextualized, informal, secret, and non-binding; they also have a short lifetime and can be updated often. Process models are highly decontextualized, formal, public, and binding, have a long lifetime, and are updated infrequently.
The **knowledge usage** part shows the relation of knowledge in the compaction process. Each more compact or decontextualized knowledge unit is based on less compact knowledge. For example, process models are based primarily on project experiences, but secondarily also on discourses, private annotations, and opinions.
In the **knowledge construction** part, the various methods of elicitation and construction of knowledge is depicted. Discourses are motivated via moderated discussions based on private annotations representing the opinions of a participant. Process experiences like lessons learned, best practices, or risk descriptions are extracted from discourses through text mining techniques, while the improved process model is generated in the process evolution step.

Experience Management

The research area dealing with support for “learning from experience” is called *Experience Management* ([6], [30]). It is concerned with the development of methods for the identification, storage, improvement, propagation, and re-/use of experiences made by 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 as an application area of Experience Management 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 an *Experience Factory* (EF) [3].

indiGo by example

indiGo’s key objective is to support the introduction and sustainable usage of process models. Sustainable means that the process models are accepted by the organization’s members, adapted to organizational changes on demand, and continuously enriched with experience from the operating business of the organization. It reaches these goals by offering members of an organization the opportunity to engage in moderated discourses about process models, and presents process-related lessons learned, that fit the current project context. Completed discourses and comments are analyzed and summarized to improve related process models and capture lessons learned from the participants. For better comprehension of indiGo, the next section is dedicated to an exemplary application of indiGo.

Imagine a senior project manager who is responsible for maintaining the process model for the process “project acquisition”. He can use indiGo in two ways: to develop and introduce the process, or to evolve it during its application. A third phase as described in a later section can be used to analyze the previous process to elicit the problems of the old process and to motivate a new one.

In the **introduction phase,** he can either rework an already existing process model or create a new one from scratch. Let’s assume he created a process model based on his own opinions and asked a colleague to check the result. With indiGo, he has the opportunity to extract experiences (like known organizational problems) from an experience factory [3] [2] while developing the model, and publish it to gain feedback from many process users.

The first step for him and the moderator is to define several goals for the discussion like: “Should the payment method be made more explicit, or should every project manager negotiate his own payment method with the customer?” Subsequently, he publishes the process model on the Intranet and invites some process users. Thereupon every participant inspects the process model based on the given goals to understand, comment, and enrich it with his/her own experiences. Simultaneously, participants look for typing errors, evaluate the ease of use, or make a dry-run of the process. Each participant can attach private annotations to the process and discuss it with other participants. The moderator summarizes the discourse from time to time and extracts ideas and lessons learned assisted by text mining techniques.

Finally, the project manager uses the ideas and experiences from the discussions to rework the process model. After the finalization of the process model, he publishes it on the corporate Intranet and informs all concerned parties of the new process model.

In the **evolution phase**, the goals of the discussion are basically the same ? keep the running process, fix minor problems, and collect experiences and opinions about the application of the process model. The model will not be subject to major revisions as it is in use at that time.

For example, assume that Ms. Legrelle (another Project Manager) has to compose an offer for a subcontract from a small start-up company. The project acquisition process has a subprocess devoted to the contract. It suggests that payment should not be performed too frequently in order to minimize administrative overhead. Ms. Legrelle feels uncomfortable with this guideline. The year before she had a subcontract with another start-up, MoCom, which went bankrupt, so that the last payment was lost, although work had been completed. Ms. Legrelle prefers to design the new offer with a frequent payment schedule, at the cost of more overheads in the administrative unit. Clearly, she should not modify the organization's process model for industrial project acquisition on her own. She would probably attach a personal note to the subprocess and initiate that her experience is recorded as a lesson learned and shared with her colleagues through the discussion forum.
Either way, if a new solution or conclusion turns up and is approved, it may be added as a new experience to the experience base. The process model would be improved periodically as substantial feedback is accumulated from the discussions and new experiences.

METHODOLOGY OF INDIGO

How an organization can accomplish process evolution using the indiGo system is the core of the process learning method. As depicted in Figure 1.3, indiGo's methodology for process learning encapsulates the several methods. This modularization enables users of indiGo to exchange these modules. For example, the eModeration method could be replaced by an "eChatting" method, or the Process Learning method by a "Software Defect Learning" method. In detail, these component methods developed in the indiGo project are:

- The **eModeration** method as a structured guideline to design, conduct, and close an eDiscussion.
- The **Text Mining** method, which supports the extraction of lessons learned, summaries, and metadata from the contributions in order to support the eModerator, process users, and authors.
- The **Process Evolution** method as the core method to enrich the process models with information from the eDiscussion in order to evolve the model.

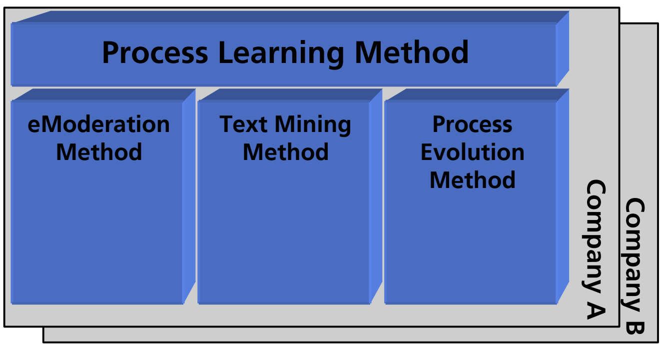

Figure 1.3 Structure of methods in indiGo

All the methods above are embedded into the **Process Learning** method that enables an organization to setup all corresponding methods and learn about its processes as well as its evolution.

Process Learning in indiGo

The Process Learning Method guides the process learning efforts performed within an organization based on the process lifecycle depicted in Figure 1.4. In particular, it coordinates the actions performed by the eModeration, Text-Mining and Process Evolution methods. It is represented as a process model and thus itself subject to process learning. The processes in the Process Learning Method are divided into three categories: core, strategy, and support processes:

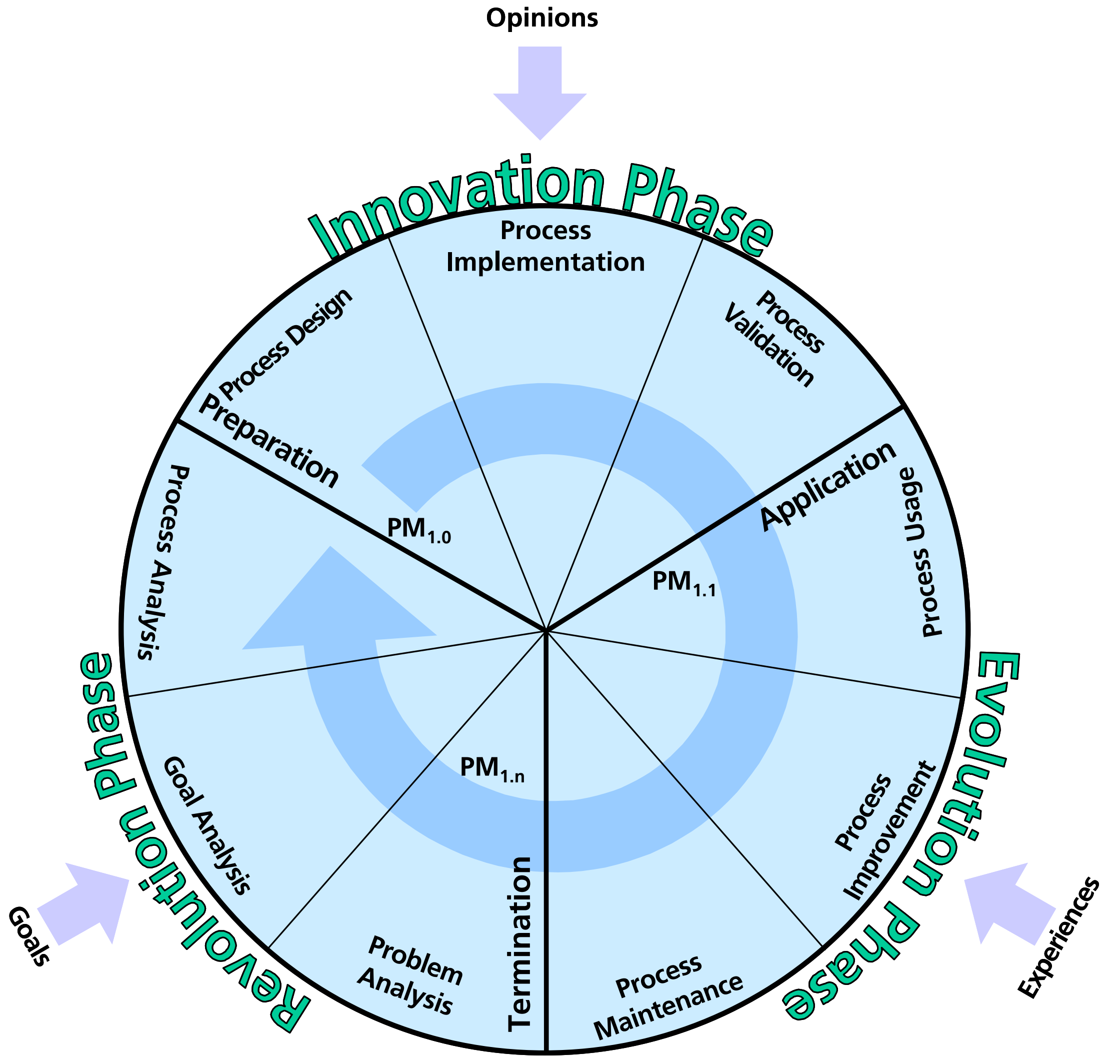

Figure 1.4 The process model lifecycle in indiGo

- **Core processes** are concerned with the creation and evolution of the process models. In the course of this process, the process learning team members interact with the organization members to create added value for the organization. As shown in the process model lifecylce depicted in Figure 1.4, these processes are divided into three phases: the innovation, evolution, and revolution phase. In the innovation phase of a process model, the process is described and subjected to "theoretical" discussion. In the evolution phase, the reworked process model is applied, and experiences about process execution are recorded. Finally, in the revolution phase, if severe problems or organizational changes occur, information about the reworking of a process model is collected, which serves as input for the next innovation phase. One example is the creation of a new process and its subsequent discussion.

- Strategy processes deal with the financing of and goal-setting for process learning. The members of the organization responsible for process learning interact with the management of the organization. Examples for such process are planning the build-up and maintenance of the process models used by the organization, or which amount of problems triggers reworking of a process.
- Support processes describe the internal actions performed by the process learning team members. An important example of such processes is the feedback session performed at the end of the introduction phase, which is aimed at the improvement of the process learning processes.

eModeration in indiGo

eModeration is the core process to keep the eDiscussions running in order to focus the discourse on the predefined goals and elicit experiences from the participants.

The eModeration process starts when the process author is ready with the first approved draft of the process model and assigns the eModerator. As input, the eModerator receives the process model and context information about the why, who, how, for whom, and for what the process is created or changed.

The eModeration process itself is split into the following steps. Steps four and five are repeated as the discourse moves onward.

1. **Design the discussion**: The frame of the eDiscussion must first be designed in cooperation with the process author. Thereby goals, topics, timeframes, environments, technologies, and target groups of the discussion are considered and fixed.
2. **Briefing & Warm-up**: The eDiscussion is prepared, participants are registered in the indiGo system and invited to the discussions. At the same time, the technical infrastructure is set up. The participants are briefed on how to use the technical system, how to correspond with other participants, as well as on the role and tasks of the eModerator.
3. **Start & Stimulate the discussion**: The beginning of an eDiscussion is a delicate process. Participants have to be informed of the discussion goals and be motivated to contribute to the discussion. To facilitate this start and direct the discussion right from the beginning, the eModerator enters contributions that stimulate discussions, such as questions concerning understandability.
4. **Control the discussion & keep it running**: If a discussion is running in the right direction and the participants are actively contributing, the eModerator has little work. In the case of a dying discussion, i.e., in case of very few contributions, the eModerator has to motivate the participants and even contribute his/her own ideas, questions, or answers. In some cases, it is appropriate to restart a discussion, include new participants, or refocus the questions and goals of the discussion.
5. **Update & Summarize contributions** to the discussion: The discussions have to be summarized and revised from time to time in order to keep the discussion on track and inform new members about previous threads. Therefore, the eModerator uses the Text Mining tools to identify experiences in the contributions and improvement suggestions. If the information is still not detailed enough, he/she can contact the author of this information directly.

6. **Close the discussion and deduce results**: At the end of the timeframe, or if the goals of the discussion are achieved, the discussion stops. In this last phase, that the eModerator has to summarize the discussion one last time, and the participants may be asked about their opinions on the discussed topics, the other participants, and the indiGo system itself. Suggestions and opinions about indiGo are forwarded to the indiGo team.
7. **Analysis and Postprocessing** of the discussion: Improvement suggestions, experiences, and other results from the discussion are prepared for the process author.

The process stops if the predefined time period ends or the goal is reached. The result of this process is then packaged as described in step 7. Furthermore, experiences elicited in the discussion are transferred into an experience database.

Text Mining in indiGo

Text mining is concerned with the task of extracting relevant information from natural language text and searching for interesting relationships between the extracted entities. From a linguistic viewpoint, natural language exhibits complex structures on different hierarchical levels, which are interconnected to each other [16]. These structures, however, are tuned to human cognitive abilities. From the perspective of a computational system, which is adopted here, linguistic information appears to be implicitly encoded in an unstructured way and presents a challenge for automatic text processing.

Text classification is one of the basic techniques in the area of text mining. It means that text documents are filtered into a set of content categories. For the task of text classification, there are promising approaches, which stand for different learning paradigms. Among them, support vector machines (SVM) are one of the most promising solutions [17]. Fraunhofer AIS has successfully applied SVM to different classification problems — topic detection and author identification [20], multi-class classification [19] — on different linguistic corpora: Reuters newswire, English and German newspapers [23], as well as radio broadcasts [13].

Text clustering is a technique to identify groups (i.e., clusters) of similar text documents. These documents are classified into one or more categories based on a pre-defined ontology or a vector containing words and phrases. While classification typically uses non-characterized documents (i.e., documents without metadata) to sort them into given categories, clustering uses characterized documents to find new or additional categories. Clusters can indicate a connection between apparently unconnected documents and help to find new groups and information as well as minimize the candidates for text-based processes.

The text summarization technique in its primitive form is used to identify meaningful and meaningless words, phrases and sentences to compose a condensed version of the text document. Summarization and rewriting of the content of a text document is very complex and has not been achieved.

The usage of other techniques of data and text mining, such as association rules, is currently not planned. Association rules could be used to find rules between contributions, thread progressions, and final results. For example, based on the first contributions in a discussion, the moderator could be informed to intensify the motivation of the participants in order to focus the discussion on the defined goals.

Process evolution in indiGo

The Process Evolution Method is concerned that changes in the process models are implemented, communicated, and recorded. The main trigger fr the actions described in the process evolution method are the improvement suggestions taken from the discussions during the innovation phase of the process. Besides adapting the process model, the evolution method describes change propagation, change information, and process model versioning. The result of an execution of the Process Evolution method is a new, published, official process model that is known to all involved members of an organization.

TECHNOLOGY OF INDIGO

The indiGo technical infrastructure consists of the groupware tool Zeno [15], the experience management environment INTERESTS [2], a tool for process modeling and publishing called Spearmint [3], as well as tools for text mining of discourses from AIS.

As shown in Figure 1.5, the indiGo platform, as installed at Fraunhofer IESE since mid 2002, integrates two independent types of systems to provide integrated process-related services to a user. While one system acts as a source for documents, like descriptions of process models, the other acts as a source for related information, like private annotations, public comments, or lessons learned from an experience base. Currently, the business process model repository **CoIN-IQ** acts as the document source, related information is provided either by **Zeno** or the lessons learned repository, **CoIN-EF**.

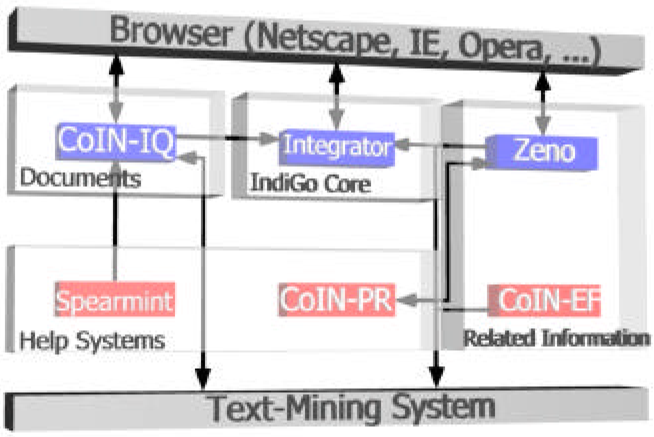

Figure 1.5 Information flow in the indiGo platform

To enhance the functionality of indiGo, we connected Zeno with **CoIN-PR** (CoIN Project Registry), a project information repository that stores all data about the projects and associated users. CoIN-PR delivers information about a specific user's current projects, which is used to index contributions in Zeno and to construct queries for CoIN-EF. To support and enhance the various roles in indiGo, **text mining tools** analyze the discussions, especially with the goal of extending or improving the lessons learned and the process models. The **Integrator** is the glue between a document server like CoIN-IQ and a server for related information like Zeno. It provides an integrated view upon the document and its related information.

Figure 1.6 Split view with CoIN-IQ at the top and a related discussion in Zeno beneath

Spearmint is IESE’s process modeling environment [3][9]. A Spearmint process model can be published on the Web as an electronic process guide (EPG) [18]. In the course of this transformation, relationships such as product flow, role assignment, or refinement are converted into hyperlinks, and the information described in the attributes appears as text in the EPG. To customize EPGs, the attributes to be generated can be specified. If a process model has been modified, the EPG can be regenerated easily. The process models in CoIN-IQ are an instance of such an EPG.

CoIN-IQ : The Process Model Repository

CoIN-IQ is IESE’s business process model repository. The topics of the processes 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).

A process within IQ is structured into "actions and subprocesses", "when to apply", "objectives, results, and quality measures", “roles involved”, “templates”, “checklists”, and "guidelines". “Actions and Subprocesses” describe the steps of the process execution. "When to Apply" gives a short overview of a process context, thus helping the user to determine wether the current process description is the desired one.

“Objectives, Results and Quality Measures” is information intended to guide the execution of a process. “Roles involved” and “Templates” provide an overview of the roles involved and templates needed in the process. These overviews are intended to support IESE members who are accustomed to the process and need a quick overview of the process. “Checklists” is also intended for the experienced user. It summarizes important steps and results of the Process Description. “Guidelines” give hints on 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.

Overview pages support a user in navigating through the processes and their elements. As process descriptions are not intended to be read on a daily basis, special attention is paid to raising awareness of changes. A special overview is devoted to the changes in and new additions of objects to CoIN-IQ. Furthermore, changes or new objects in CoIN-IQ are marked by a "new" or "changed" icon (see top of Figure 1.7). Finally, the most recent changes are announced on the entry page of CoIN-IQ.

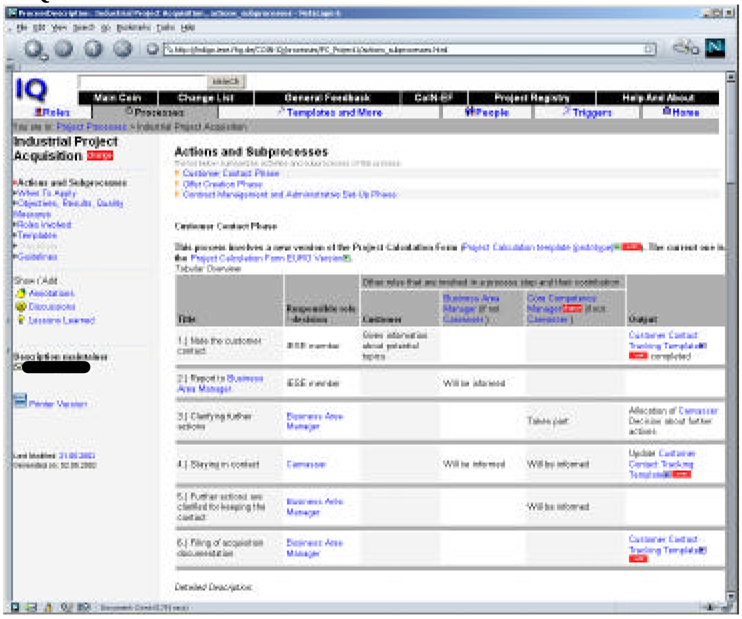

Figure 1.7 Screenshot of a process description.

Zeno: The eDiscussion Groupware

The objective and major concept of Zeno is to provide its users with a tool for structured and distributed discussions.

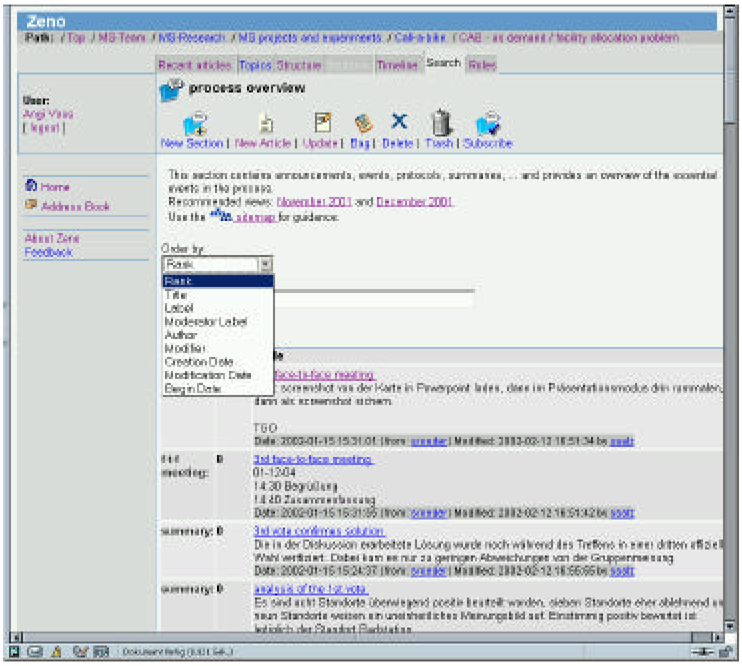

Figure 1.8 Screenshot of the Zeno discussion overview

It was first presented at CeBIT 1996 and continuously improved up to version 1.9 in 1999. Since then, a completely new system has been realized, which addresses a broader spectrum of discourses in the knowledge society: participatory problem solving, consensus building [33], mediated conflict resolution [24], teaching and consulting. The new Zeno focuses on eDiscourses and supports eModerators in turning discussions into discourses, elaborating the argumentation and carving out rationales.

A discourse is a deliberative, reasoned communication; it is focused and intended to culminate in decision making [14]. Turoff et al. [31] argued that building a discourse grammar, which allows individuals to classify their contributions into meaningful categories, is a collaborative effort, and its dynamic evolution is an integral part of the discussion process. A discourse grammar (or ontology) defines labels for contributions, labels for references (directed links) between contributions, and may constrain links with respect to their sources and targets. Supporting communities in evolving their own discourse grammars has been a key issue in the design of Zeno.

CoIN-EF : The Lessons Learned Repository

The lessons learned repository CoIN-EF captures experience about project management from IESE. Before retrieving lessons learned via indiGo, these lessons learned were captured by project retrospectives. To search for a lessons learned applicable to the current project of a user, the user describes the context of this project (e.g., project type, topics). CoIN-EF compares this context to the project context of the available lessons learned with similarity-based retrieval [22]: A measure indicating the degree of similarity in the current context is calculated. The lessons learned are ordered according to this similarity value. Queries can be entered in the query interface shown in Figure 1.9 or via indiGo. When CoIN-EF is queried via indiGo, results from CoIN-EF can be retrieved without further user interaction, since the context of the current project of a user is already captured in the preferences.

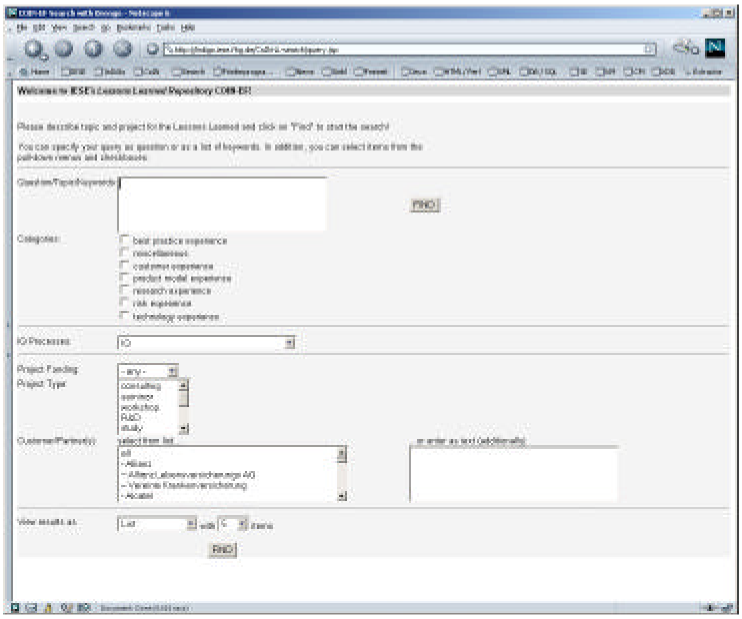

Figure 1.9 Screenshot of CoIN-EF

The *lessons learned* can cover different topics and take on different forms [7]. Within CoIN, lessons learned about project management are captured. One lesson learned can take on the form of an *observation*, *a problem*, *a guideline*, *a pragmatic solution*, or an *improvement suggestion*. Each lesson learned is personalized to allow a querying user to ask a colleague for further information. The context of these lessons learned is 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 lesson learned was gained. Therefore, a project team member can specify his/her current environment as well as the current situation to search for similar experiences. Figure 1.10 shows the interrelations between the context and the different types of lessons learned. Numbers represent the cardinality of the relationship between the context or different types of lessons learned.

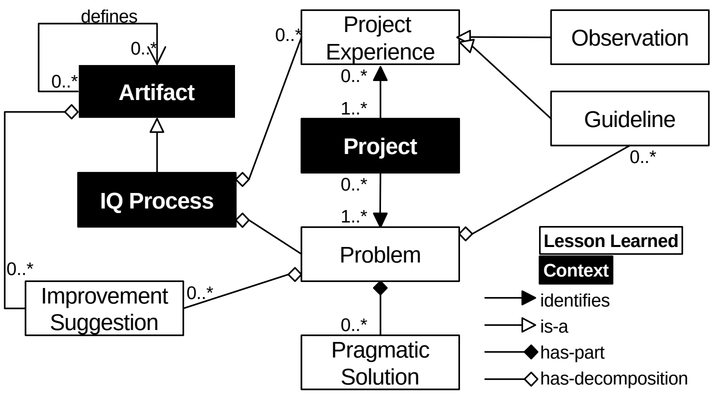

Figure 1.10 COIN Ontology according to Tautz [30]

- *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. E.g., 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.

RELATED WORK

One central issue in knowledge management is to offer the right knowledge at the right time. As the domain of indiGo is based on process models, they form the backbone for knowledge delivery. While applying a particular process model, members of the organization find supplementary knowledge with regard to the users' current project context. This supplementary knowledge is provided through associated discussions in the users' groups, private annotations, and of course, records lessons learned from other projects. In the remainder of this section, we discuss several related systems for participative process learning as realized by the indiGo approach.

As a preliminary conclusion, the platform and methodology of indiGo is more comprehensive than other approaches to organizational process learning ([30], [6], [27]) and distributed knowledge management. indiGo bridges the gap between informal, communication-oriented knowledge and formal, organization-oriented knowledge and provides a socio-technical solution that covers individual knowledge usage as well as social knowledge creation.

In Dellen, Könnecker, and Scott [9], relevant process modeling editors and publication software are summarized. From the perspective of process learning, three kinds of tools can be distinguished:

(a) Software that publishes the process model in a representation that is understandable to humans,

(b) Software that, additionally, allows a user to annotate or to discuss process models, and

(c) Software that focuses on the collaborative creation of process models, that is, on how process engineers and authors can create and manipulate process models.

While (a) is a passive way of communicating process models that have to be complemented by organizational measures to induce real change, (b) allows two-way communication between the process engineer or the author and organizational members. (c) concentrates on supporting process engineers and authors in the creation of the process models, which in practice will also include discussions.

For each of those categories, Table 1.1 gives some examples. The Process Model (No 1) belongs to category (a). It is focused on business process design and improvement of ISO 9000 processes. For category (b), a prototype extension of Spearmint was developed to gain some first experiences with annotations and discussion on a private, groupwise, and public level (No 3).

Furthermore, PageSeeder can be used to augment the HTML representation generated from the process modes (EPG) [28] (No 4). DaimlerChrysler's LID system [32] allows public annotation of software process models, which the process engineer can distill into lessons learned and attach to the process model (No 5). Finally, as representatives of category (c), ADONIS (No 6) and ARIS Web Designer (No 10) focus on collaborative editing and publishing of graphically represented business process models. ARIS also offers support for enacting the business process models, for instance, via Lotus Notes.

Name	Publi-cation	Annot-ation	Discus-sion	Coll. Creation	URL / further information
Process Model	X				www.processmodel.com
Process	X				www.scitor.com/pv3/purchase.proces.asp
SPEARMINT / Annotation	X	X	X		www.iese.fhg.de/Spearmint_EPG/
SPEARMINT / PageSeeder	X	X	X		www.iese.fhg.de/Spearmint_EPG/
LID System	X	X	X		[32]
ADONIS	X			X	www.boc-eu.com
INCOME	X			X	www.promatis.de
INNOVATOR	X			X	www.mid.de
in-Step	X			X	www.microtool.de
Aris: Web Designer	X			X	www.ids-scheer.de
Aris: Web Publisher	X				www.ids-scheer.de

Table 1.1 Overview of process modeling and publication software

EVALUATION OF INDIGO

The methodology and technical system developed for indiGo were evaluated through a case study, which was performed at the Fraunhofer Institute for Experimental Software Engineering (IESE) starting in the summer of 2002. The framework of the evaluation of the indiGo approach is described in the following section. First results from the evaluation are given in the subsequent sections. A more detailed description of the case study and its results is available under [10].

Case study context & design

The process models examined describe the "Industrial Project Acquisition" (IPA) and "Conference Participation Planning" (CPP) processes. The main objective of the case study was to show that the participative approach to process introduction in an organization improves the quality of the process model as well as its acceptance. The secondary objective was to collect practical experience for further improvement of the indiGo technical infrastructure. In this section, the context of the case study is described first, followed by its design and the tools used for evaluation.

The *IESE as setting of the case study* employed about 97 full-time employees at the time of the case study. Of these, 70 scientists worked in applied research as well as the evaluation and transfer of software engineering knowledge in a broad range of industrial and publicly funded projects. Fraunhofer IESE has two spatially separated offices, so indiGo offered a convenient opportunity for employees of both offices to participate in the discussion.

As projects are the core business of IESE, process models are central and affect every employee. It is vital that employees accept and "live" the process models and cooperate to continuously improve them. Due to the variety of projects, the processes can reasonably be captured at an abstract and a decontextual level only. That means, the performance of the application of an abstract process model in real life is very knowledge-intensive.

The task to perform was the creation of new as well as the adaptation of existing process models due to a recent restructuring of IESE. Process models with a high potential of uncertainty and conflicting interpretations were selected for the case study. Persons concerned with organizational topics were chosen as owners of the respective process models. Every scientific member of IESE had the opportunity to participate in the case study and was given a short training on the indiGo system.

The case study itself had a pre-post design. At the start of the discussion in June 2002, a questionnaire was distributed among the IESE members to get personal evaluations of the quality and acceptance of those process models. After the improvement suggestions resulting from the discussions were implemented, a similar questionnaire was distributed to evaluate the changed process in July 2002.

Case study results

The presentation of the case study results is divided into three parts: First, the actual participation of IESE members is presented. Second, the differences in acceptance and perceived quality are presented. Third, the main practical experiences and findings are presented.

The actual participation is presented in Table 1.2. The differences in acceptance and perceived quality presented in the next section are based on those 16% of all IESE members who completed both questionnaires. None of the participants who completed the first and second questionnaire were part of the project members of indiGo. Since the absolute number of participation is quite small, transferring these results to other organizations should be done with caution. Based on the case study, further evaluations will be performed at IESE and in future projects. Nevertheless, the results of this case study indicate that the effects observed can be replicated in these future evaluations. (Threats to validity are discussed in detail in [10].)

Participant in	Participants	» % (from 97)
1st Questionnaire	24	25 %
2nd Questionnaire	26	27 %
1st and 2nd Questionnaire	15	16 %
Discussion	21	22 %

Table 1.2 Distribution of participants

For measuring *acceptance and perceived quality* (single quality aspects and overall quality aspects), two major findings hold for both processes: When the results of the pre-phase (1st questionnaire) are compared to the ones in the post-phase (2nd questionnaire), the median of all results improves. The only exception is the median of acceptance for CPP, which remains stable. Furthermore, the bandwidth of results decreases, i.e., participants evaluate the process in the pre-phase more differently than in the post-phase. In other words, the resulting processes are evaluated better and more consistently with respect to acceptance and perceived quality.

These effects are depicted exemplarily by the results of IPA in Figure 1.11. For the single quality aspect measure shown, the median increased from about 0.77 to 0.90 (with 1.0 being the best possible result for this measure). The overall quality aspect measure increased from about 0.8 to 0.83 (again, 1.0 being the best possible result). As depicted, the median of acceptance measurement increased 2 to 1 (with 1 being the best and 6 being the worst measure). The significance of the difference – i.e., whether the difference is caused coincidentally or has a statistical significance – was investigated using the Wilcoxon matched pair test [29]. For case studies like these, a level of significance or P-value of 10 % or lower [4] is an acceptable indicator of significance. The P-values for this test are given in Table 4, where N indicates the number of participants who provided data for measurement. Values where the test was successful (i.e., the P-Value is below < 0.1) are formatted in bold.

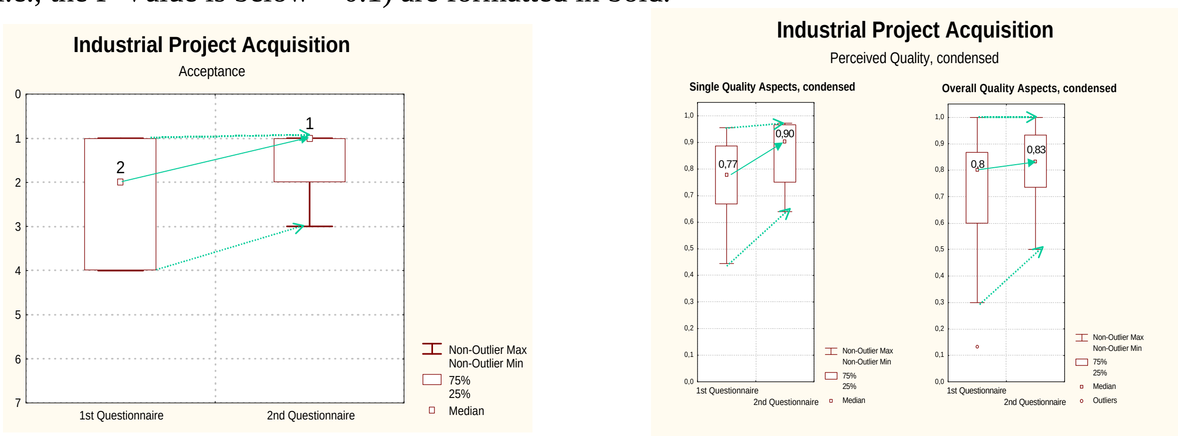

Figure 1.11 Pre-post evaluation of perceived quality for Industrial Project Acquisition

The Wilcoxon matched pair test was successful for two of the three criteria of each process. The criterion where it failed differed between the two processes: The test for Overall Quality Aspects failed for the IPA process. For the CPP process, the test failed for the Acceptance aspect. Therefore, the improvement observed has to be checked in future evaluations especially for these aspects with failed tests. Furthermore, due to the low number of participants, the power could not be calculated and no statement can be made on whether no difference is, in fact, present. (For details, refer to [10].)

Criteria	N	P-Value: Industrial Project Acquisition	N	P-Value: Conference Participation Planning
Acceptance	10	**0.051922**	12	0.767099
Single Quality Aspects	14	**0.009637**	15	**0.030335**
Overall Quality Aspects	14	0.401684	15	**0.074745**

Table 1.3 P-Values of significance

The decreasing result bandwidth is shown graphically by smaller boxes (25% - 75%) and the distance between the non-outlier min and non-outlier max (see legend for details) between both phases.

The practical experiences gathered regarding the indiGo technical infrastructure, eParticipative Process Learning in general, and the eModeration method in particular add to the above findings:

- For the technical infrastructure, discussion groups about indiGo itself were the most important source of improvement suggestions. From 36 contributions, 26 improvement suggestions could be deduced.
- Concerning eParticipative Process Learning, 25 improvement suggestions could be deduced from 120 contributions in four weeks. 15 of them were implemented. The first questionnaire revealed a generally positive attitude towards process discussions and experience sharing. For the process modeling team, the possibility to delegate open questions to the discussion accelerated modeling.

- The eModeration method was improved by several lessons learned from the case study. For example, the same person should not perform the role of the eModerator and Process Author as it was done during the case study. The reasons for this separation are potential and assumed conflicts of interests between these roles. Furthermore, most of the participants in the 2nd questionnaire were satisfied with the relevance, results, and moderation of the discussions.

Simplified, the case study showed the following: acceptance and perceived quality increase with process discussion, and indiGo supports this discussion well. Due to the (potential) involvement of all organizational members, improvement suggestions concerning the processes could be collected that would not have been (practically) collected in classical, workshop-based process modeling.

SUMMARY & OUTLOOK

indiGo has proven to be a valuable system for a globally distributed organization for learning and improving their process models. It enables a distributed organization to identify and record experiences from participants of discussions in order to feed them back into an organization-wide experience base. indiGo is designed to support all kinds of knowledge that has been identified as being important for organizational process learning. Applied in a distributed environment, it can be used at the same time for distributed inspection (i.e., eProcess Inspection) or distributed evolution (i.e., eProcess Evolution) of process models.

The first version of indiGo was presented in March 2002 at CeBIT. Starting May 2002, indiGo was evaluated within a case study. New project and strategy processes were introduced for the organization, and indiGo was chosen as the organizational process learning platform. Besides improving the discussed process models, we received valuable feedback for all the described methods and technologies of indiGo.

As the indiGo project ended with the year 2003, the ideas, methods, and tools will be used in future projects. The new project RISE (Reuse in Software Engineering; funded by the German Ministry of Education and Research, BMBF) will focus on the distributed management of knowledge about software defects and the removal of these defects by distributed teams.

ACKNOWLEDGEMENTS

indiGo (Integrative Software Engineering using Discourse-Supporting Groupware) is funded by the German Ministry for Education and Research ("Bundesministerium für Bildung und Forschung") (BMBF) under grant number 01 AK 915 A [see http://indigo.fhg.de/].

Thanks goes to Astrid Haas for her work on eModeration and Torsten Willrich as well as Markus Nick for their help in the development of indiGo. We also thank our students for their work on the first prototypes of the indiGo technical architecture and process model processing framework.

REFERENCES

[1] Althoff, K.-D., Becker-Kornstaedt, U., Decker, B., Klotz, A., Leopold, E., Rech, J. & Voss, A. (2002) The indiGo Project: Enhancement of Experience Management and Process Learning with Moderated Discourses In P. Perner (Ed.), Data Mining in Marketing and Medicine, Springer Verlag, LNCS, 27 pages

[2] Althoff, K.-D., Decker, B., Hartkopf, S., Jedlitschka, A., Nick, M. & Rech, J. (2001). Experience Management: The Fraunhofer IESE Experience Factory. In P. Perner (ed.), Proc. Industrial Conference Data Mining, Leipzig, 24.-25. July 2001, Institut für Bildverarbeitung und angewandte Informatik

[3] 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.

[4] Briand, L.; Bunse, C.; Daly, J: A Controlled Experiment for Evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs, ISERN Technical Report, 99-07, Kaiserslautern, 1999

[5] Ulrike Becker-Kornstaedt, Dirk Hamann, Ralf Kempkens, Peter Rösch, Martin Verlage, Richard Webby, and Jörg Zettel. Support for the Process Engineer: The Spearmint Approach to Software Process Definition and Process Guidance. In Proceedings of the 11th Conference on Advanced Information Systems Engineering (CAISE '99), Heidelberg, Germany, June 1999. Lecture Notes on Computer Science, Springer-Verlag.

[6] Bergmann, R. (2001). Experience management — foundations, development methodology, and internet-based applications. Postdoctoral thesis (Habilitations-Schrift), Department of Computer Science, University of Kaiserslautern.

[7] Birk, A.; Tautz, C. (1998) 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.

[8] Bußmann, Hadumod (1990). Lexikon der Sprachwissenschaft, Kröner: Stuttgart.

[9] Raimund L. Feldmann, Michael Frey, Marco Habetz, and Manoel Mendonca (2000). Applying Roles in Reuse Repositories. In Proceedings of the Twelfth Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, July 2000. Knowledge Systems Institute, Skokie, Illinois, USA.

[10] Decker, B.: IndiGo: Fallstudie Prozesseinführung. (in German) indiGo Fallstudie http://indigo.fraunhofer.de, 2003 Link checked on 28.04.03.

[11] Dellen, B., Könnecker, A. & Scott, L. (2000). Process Modeling Tool Evaluation. IESE-Report 002.00/E, Fraunhofer IESE, Kaiserslautern, Germany.

[12] Dingsøyr, Thorgeir (2001) Classification of collected experience in CoIN Internal Report, Fraunhofer IESE, 22.3.2001

[13] Eickeler, Stefan & Kindermann, Jörg & Larson, Martha & Leopold, Edda & Paaß, Gerhard: Classification of Spoken Documents using SVM and Subword Units; submitted to Neurocomputing, Special Issue on Support Vector Machines.

[14] Erickson, T. (1999): "Persistent Conversation. An introduction". Journal of Computer-Mediated Communication, 4.

[15] Gordon, Thomas F., und Nikos Karacapilidis. 1997. "The Zeno Argumentation Framework." Pp. 10-18 in Proceedings of the Sixth International Conference on Artificial Intelligence and Law.

[16] Hrebícceck, Ludek & Altmann, Gabriel (1996): The Levels of Order in Language, Glottometrika 15, 38-61.
[17] Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. Proc. 10th European Conference on Machine Learning (ECML 98), LNCS 1398, Springer Verlag, pp137-142
[18] Kellner, Marc I.; Becker-Kornstaedt, Ulrike; Riddle, William E.; Tomal, Jennifer and Verlage, Martin (1998). Process Guides: Effective Guidance for Process Participants. In Proceedings of the Fifth International Conference on the Software Process, pages 11-25, Chicago, IL, USA, June 1998. ISPA Press.
[19] Kindermann, Jörg & Paaß, Gerhard & Leopold, Edda (2001): Error Correcting Codes with Optimized Kullback-Leibler Distances for Text Categorization; PKDD'01; 3 — 7 September 2001 in Freiburg.
[20] Kindermann, Jörg & Diederich, Joachim & Leopold, Edda & Paaß, Gerhard: Identifying the Author of a Text with Support Vector Machines, accepted at Applied Intelligence.
[21] Kohonen, Teuvo (2001): Self-organizing maps, Springer: Berlin, Heidelberg, New York
[22] Kolodner, J. L. (1993). Case-Based Reasoning. Morgan Kaufmann.
[23] Leopold, Edda & Kindermann, Jörg (2002): Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?; in: Machine Learning 46, 423 — 444.
[24] Märker, O., Hagedorn, H., Trénel, M. and Gordon, T. F. 2002 'Internet-based Citizen Participation in the City of Esslingen. Relevance - Moderation - Software', in M. Schrenk (ed) CORP 2002 - "Who plans Europe's future?" Wien: Selbstverlag des Instituts für EDV-gestützte Methoden in Architektur und Raumplanung der Technischen Universität Wien.
[25] Mehler, Alexander (2002): Hierarchical analysis of text similarity data, in: Thorsten Joachims & Edda Leopold (eds.): Künstliche Intelligenz 2/02, Schwerpunkt Textmining, P. 12-16.
[26] Merkl, Dieter (1997): Exploration of Document Collections with Self-Organizing Maps: A Novel Approach to Similarity Representation, a novel approach to similarity visualization". Proceedings of the European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'97), (Trondheim, Norway. June).
[27] Minor, M. & Staab, S. (eds.) (2002). 1st German Workshop on Experience management – Sharing Experiences about the Sharing of Experience, Berlin, March 7-8, 2002, Lecture Notes in Informatics, Gesellschaft für Informatik (Bonn).
[28] Scott L, Jeffery D.R., Becker-Kornstaedt U. (2001). Preliminary Results of an Industrial EPG Evaluation. In F.Maurer; B. Dellen; J. Grundy; B. Koetting (eds.), Proc. 4th ICSE Workshop on Software Engineering over the Internet, Toronto Canada, 12-19 May. 2001, IEEE Computer Society, California, USA, pp. 55-58.
[29] Sheskin: Handbook of parametric and non-parametric statistical procedures (2nd edition), 2001
[30] Tautz, C. (2000). Customizing Software Engineering Experience Management Systems to Organizational Needs. Doctoral Dissertation, Department of Computer Science, University of Kaiserslautern, Germany.
[31] Turoff, M., Hiltz, S. R., Bieber, M., Fjermestadt, J. and Ajaz, R. (1999): "Collaborative discourse structures in computer-mediated group communications". Journal of Computer-Mediated Communication, 4.

[32] von Hunnius, J.-P. (2000). WESPI – Web Supported Process Improvement. In K.-D. Althoff & W. Müller (eds.), Proc. 2nd International Workshop on Learning Software Organizations, June 20, 2000, Oulu, Finland, Fraunhofer IESE, Kaiserslautern, Germany.
[33] Voss, A., Roeder, S. and Wacker, U. (to appear (2002)): „IT-support for mediation in decision making - A role playing experiment”. In Online-Mediation. Theorie und Praxis computer-unterstützter Konfliktmittlung.(Eds, Märker, O. and Trenél, M.) Sigma, Berlin.
[34] IndiGo Web-Page: http://indigo.fraunhofer.de, last visited 27.01.04