Enhancing Experience Management and Process Learning with Moderated Discourses: the indiGo approach

Klaus-Dieter Althoff[1], Ulrike Becker-Kornstaedt[1], Björn Decker[1],
Andreas Klotz[2], Edda Leopold[2], Jörg Rech[1], Angi Voß[2]

1. Introduction

The business process models of organizations operating in the innovative software market are one of their major knowledge assets. However, these models need to be constantly evaluated and hardened in the business of those organizations and enhanced by further knowledge to make them operable. The approach of the project indiGo[3] (Integrative Software Engineering using Discourse-Supporting Groupware) is to support this evaluation and enhancement offering members of an organization to engage in discourses about the process models and their execution (communities of practice) and by presenting process-related lessons learned fitting to the current project context. On the organizational level, finished discourses will be analyzed and summarized to improve process models (process learning) and create new lessons learned (learning from experience). To achieve these objectives, indiGo will develop an integrated, comprehensive set of methods and a technical infrastructure as a joint effort of two Fraunhofer Institutes: Fraunhofer IESE (Institute for Experimental Software Engineering) in Kaiserslautern and Fraunhofer AiS (Autonomous Intelligent Systems) in Sankt Augustin.

The indiGo methods and tools create added value through the speed-up of innovation cycles by involving more people, recording more information on processes in the form of discourses, and improving the construction of organizational knowledge through the preparation and wrap-up of the discourses using text mining and case-based reasoning. The indiGo project enhances current approaches to experience management by providing a solution integrating discussion results into an experience base, a repository for the reuse of experience. The final indiGo technical infrastructure will consist of the Zeno® groupware tool of AiS, IESE's experience management environment INTERESTS, tools for process modeling and publishing, as well as tools for text mining of discourses. Both the developed methods and the indiGo architecture will be evaluated within a case study on process learning carried out at IESE.

2. indiGo – the framework

IndiGo' key objective is to create and sustain living process models, i.e., process models that

[1] Fraunhofer–Institut für Experimentelles Software Engineering (IESE), Sauerwiesen 6, 67661 Kaiserslautern, althoff@iese.fraunhofer.de

[2] Fraunhofer-Institut für Autonome Intelligente System (AIS), Schloss Birlinghoven, 53754 Sankt Augustin, angi.voss@fraunhofer.de

[3] indiGo (http://www.ais.fraunhofer.de/MS/projects/indigo/) is funded by the German Ministerium für Bildung und Forschung under grant number 01 AK 915 A

are accepted by the organizations members, adapted to organizational changes on demand and continuously enriched with experience from the operating business of the organization.

For example, assume Ms. Legrelle, a team leader in the organization, has to compose an offer for a subcontract from a small start-up. The process model for the acquisition of industrial projects has a subprocess devoted to the contract. It suggests that the payment scheme should not be too fine-grained in order to minimize administrative overhead. Ms. Legrelle feels uncomfortable with this guideline. The year before she had had a subcontract with another start-up, Orion, which got bankrupt, so that the last payment was lost for her team although they had completed the work. Ms. Legrelle prefers to design the new offer with a frequent payment schedule, at the cost of more overhead in the administrative unit.

Clearly, Ms. Legrelle would not like to modify the organization's process model (1) for industrial project acquisition on her own - it is not her job and her view may be too subjective. She would probably agree that her experience with the Orion project be recorded as a lesson to be learned, but even so, she would hardly take the trouble to fill in the required form to create an "official" case (2). Rather, she would like to suggest her exception from the guideline to her colleagues, backed up by the example of Orion, and wait for their responses (3). Whatever the conclusion, she would probably add it as a personal note (4) to the guideline in the respective subprocess.

2.1 Knowledge compaction, usage and construction

indiGo takes into account all four kinds of knowledge occurring in the example and supports them as successive stages in a process of knowledge compaction (aggregation, condensation, summarization or classification). Figure 1 arranges the four knowledge categories on one layer and embeds it into layers of knowledge usage and knowledge construction.

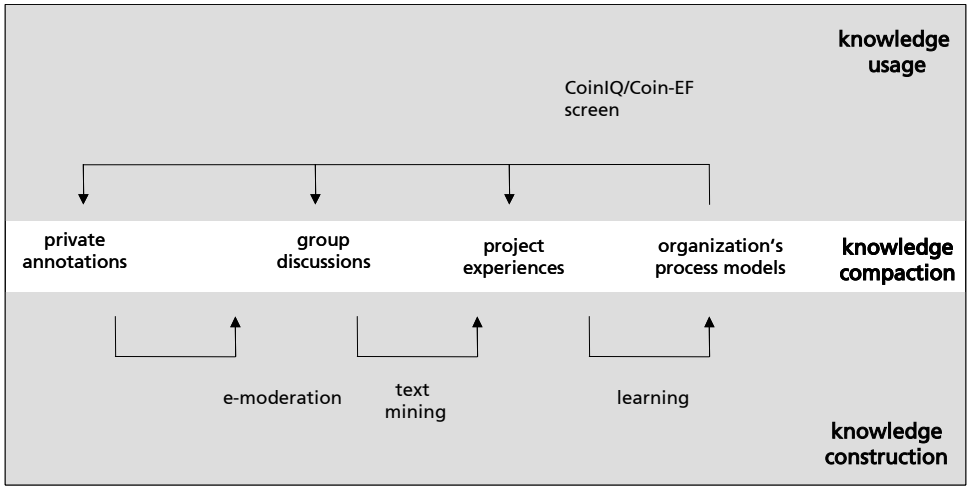

Figure 1: layers of knowledge compaction, usage and creation for process-centered applications

Knowledge compaction is a process of decontextualization (a) and formalization (b) with the goal of decreasing modification times (c) as well as increasing lifetime (d) and obligingness (e); and of course more obliging knowledge should be more visible (f). As indicators of knowledge compaction (a-f) are correlated, and they exhibit a clear progression from private annotations to group discussions, to stored cases, to an organization's process models. Pri-

vate annotations are highly contextualized, informal, secret, and non-binding, they have a short lifetime and can be updated often, while process models are highly decontextualized, formal, public, and obliging, they have a long lifetime and are updated infrequently.

The central issue in knowledge usage is how to offer the right knowledge at the right time. As the domain of indiGo is dominated by process models, they should form the backbone for knowledge delivery. While applying (instantiating) a particular process model, members of the organization should find - a mouse click away - supplementary knowledge in associated cases that are dynamically retrieved with regard to the users' current project context. The supplementary knowledge is provided through associated discussions in the users' groups and in their private annotations.

If no relevant knowledge is available, the users have encountered a gap in the knowledge. If they know a solution themselves, they may write a quick private note and attach it to the  rent part of the process model. Otherwise, they may raise the problem in one of their discussion groups. Other users may be able to help, possibly they had been confronted with a similar problem formerly and had written a private note to remember the solution. Then they may bring this note into the group discussion.

Either way, if a new solution turns up and stands its test, it may be added as a new case to the experience base. The process model would be adapted periodically as substantial feedback is accumulated from the discussions and the new experiences. Again, contributing new bits of knowledge should be a matter of very few mouse clicks.

To extract knowledge from a discussion for the experience base the indiGo system will be enhanced by text mining tools, and the experience base should offer analytic tools that cluster, categorize, or differentiate the cases as input for improving the process models.

Indigo is more comprehensive than other approaches to organizational learning (Tautz 2000, Bergmann 2001, Kluge 1999, Minor & Staab 2002) because it 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.

3. indiGo – the methodology

The social side of indiGo, its methodology, describes how an organization can accomplish process learning using the indiGo platform (its technical side).

3.1 indiGo – introduction by bootstrapping

The indiGo methodology is in itself phrased in terms of a process model, called the indiGo model for process learning. The self description of the indiGo methodology through indiGo process models offers the opportunity to 'bootstrap' indiGo, i.e., to apply indiGo to itself. First, it allows to have a test run of both the methodology and the technical infrastructure during the introduction of indiGo. Furthermore, since the persons involved in the indigo introduction directly experience the difficulties with this approach in their organization, it will be their prime interest to resolve these difficulties is within the interest of those roles and can be solved without annoying the members of the organization. Therefore, the members of the organiza-

tion can rely on a tested infrastructure and a consolidated team to support them in the roll-out phase.

The bootstrap approach to introducing indiGo also implements three feedback cycles: a process-related, an organization-specific-methodic, and a general-methodic one. The process-related feedback cycle is the application of the indiGo methodology to the processes of the organization. The organization-specific-methodic one is the continuous improvement of the indigo methodology at a specific organization. The general methodic one is the feedback of experience gained by introducing the indiGo methodology by the supporting organizations, thus improving the generic process model of indiGo.

3.2 Process learning processes

The processes of the indiGo model are ordered into three groups: core processes, strategy processes and support processes. Core processes generate a direct benefit for the organization: creation of a process model, introduction of a process model, supporting process execution, and maintenance of a process model. Strategy processes cover the strategy definition of the experience factory (EF) (Basili, Caldiera & Rombach 1994). These strategy processes include the definition or update of subject areas, setting objectives for the subject areas, and creating a short and long term perspective for the experience factory. Support processes support core processes or strategy processes and include moderating discussions, processing lessons learned based on contributions to the discussion, handling feedback, managing the experience factory, and defining requirements for improving the technical infrastructure.

An instantiation of the indiGo methodology will be performed as follows: First, subject areas are defined and prioritized. The prioritization is used to select subject areas for the test run, the roll-out phase and future opportunities to enlarge the scope of process learning. Second, organization members are assigned to the roles and subject areas. Third, the generic process model of indiGo is instantiated to the needs of the organization by discussing them via the indiGo technical infrastructure. This discussions are continued throughout the application of the indiGo methodology, thus adapting and improving the processes of the indiGo methodology.

3.3 Role Model

The indiGo role model and subject areas together build a fine-grained framework that allows to adapt the indiGo methodology to the needs and settings of the organization. Since the subject areas are organization-specific, they will not be detailed further. However, for the role model a description can be given at a generic level.

- The task of the members of the organization is to discuss the process models, report problems in process execution and provide experience relevant for the processes. These contributions are further processed by the members of the experience factory team.
- The moderator facilitates the discussion of the members of the organization. He / she holds close contact to the process owner and authors to start discussions with relevant topics. From time to time, the moderator summarizes the discussion to help new organizational members to catch-up with the discussion. In the end of a discussion, the moderator also creates a summary for the EF Team and the process owner.

- The process owner is responsible for a set of processes, often about a certain subject area. Due to his/her position within the organization, the process owner is allowed to take decisions about the definition and content of a process. Examples for such positions are the upper management for core processes of an organization or the provider of a certain service for support processes.
- The process author is responsible for creating and maintaining process descriptions as a whole or parts of it. If not performed by the same person, the process author supports the process owner by preparing decisions of the process owner.

These roles are supported by the EF team, which comprises a process engineer, experience manager, project supporter, and librarian (Feldmann, Frey et al. 2000). It is possible to assign several roles in the EF team to one person, thus lowering the dedicated resources. Furthermore, responsibilities for process learning activities are assigned to organization members outside the EF team that create synergies for those organizational members. This also lowers the need for dedicated resources and creates acceptance for the process learning activities.

3.4 Case study – indiGo for business process

The methodology and software developed for indiGo will be evaluated through a case study, which will be performed at Fraunhofer IESE starting in April 2002. First results should be available at the workshop.

IESE's approximately 100 regular staff do applied research, evaluation and transfer of software engineering methods and techniques in a broad range of industrial and publicly funded projects. Organizational knowledge management is delegated to the CoIN-Team (CoIN = Corporate Information Network) with its five part-time members. They maintain the process model database (CoIN-IQ) and a database of experiences with current and completed projects (CoIN-EF). The process models are partitioned into subject areas, for instance project-related matters are distinguished from cooperation with universities, and persons concerned with the subjects in the organization are selected as owners of the respective process models. Lessons learned from the projects are elicited by the CoIN-team which also provides the process engineer.

The process models concerned with project management need to be adapted to a recent restructuring of the organization. As projects are the core business of IESE, the new process models are central for the organization and affect most of the staff. It is vital that they accept and "live" the new process models and cooperate to continuously improving them. Due to the variety of the projects, the processes can reasonably be captured at an abstract level only. That means, the instantiation of the abstract process models is highly knowledge-intensive.

In a series of workshops, which involved the higher management, an initial revision of the process descriptions was elaborated. Through regular informal contacts it was assured that the higher management would support the introduction of the new processes. Process models with a high potential of conflicts will be introduced in April 2002 according to the indiGo methodology.

The process of creating project offers is planned to be introduced in two phases: a discussion phase and a pilot phase. In the discussion phase members of the organization discuss

the process description without actually instantiating it. This will elicit not only suggestions on the process descriptions, but related stories or examples from their daily work. A member of the CoIN-team or even an independent moderator will facilitate the discussion. The author of the process description will point out topics for leading the discussion in a goal-oriented way. The participants are asked to indicate the role of their contributions by using a set of labels that was specially designed for this discourse. The use of labels shall stimulate a rational and easier to understand argumentation. In case of need the moderator will contact experts to comment on a contribution. For example, the lawyer could answer questions on the new laws of warranty that became effective at the beginning of this year. On terminating the discussion phase, the process author and the responsible person from the CoIN-team will secure the contributions. Extracted experiences have to be approved by the responsible process owner before they are transferred to the experience base. In some cases, further focused investigations, like a project analysis, will be taken into consideration (Tautz 2000). Text mining methods for clustering contributions or adding semantic links will enhance the analysis conducted by both, the process author and the experience engineer. Selected contributions, especially open arguments, will remain with the process model.

In the subsequent pilot phase the process descriptions will be evaluated at daily work. Now practical problems will turn up that have to be solved by the staff or some experts. These discussions, too, will be evaluated for improving the process descriptions and extending the experience base. Clearly, the emphasis will now lie on gathering experiences while the process model will stabilize.

Finally, the revised process description will be published for regular operation. Due to the comprehensive validation during the discussion and pilot phases the number of new contributions is expected to be small. Therefore, the responsible process author and his proxy may monitor the discussion with low effort beside their other tasks. For the process of writing project offers, this would be the administrative person whose work will be alleviated through efficient instantiations of the process model. Supported by text mining services the responsible moderator will continue to observe the discussions in order to identify interesting experiences and projects that require further analysis.

If by and by problems of executing the process models are starting to accumulate a new revision will be scheduled.

4. indiGo – the software platform

The indiGo technical platform integrates two independent types of systems for a completely new service. While one system acts as a source for documents, like descriptions of business process models, the other acts as a source for related information, like private annotations, public comments or lessons and examples from an experience base. The business process model repository CoIN-IQ acts as the document source, related information is provided by the groupware Zeno or the experience management system CoIN-EF.

As shown in figure 2 the indiGo platform, as presented at CeBIT 2002, consists of three core components. The integrator acts as a middleware between the document and information source. On the left hand side CoIN-IQ, as the document source, hosts the business process models that can be supported by the information from the second system. Zeno, as the information source on the right side, manages annotations and discussions about the business process models from CoIN-IQ.

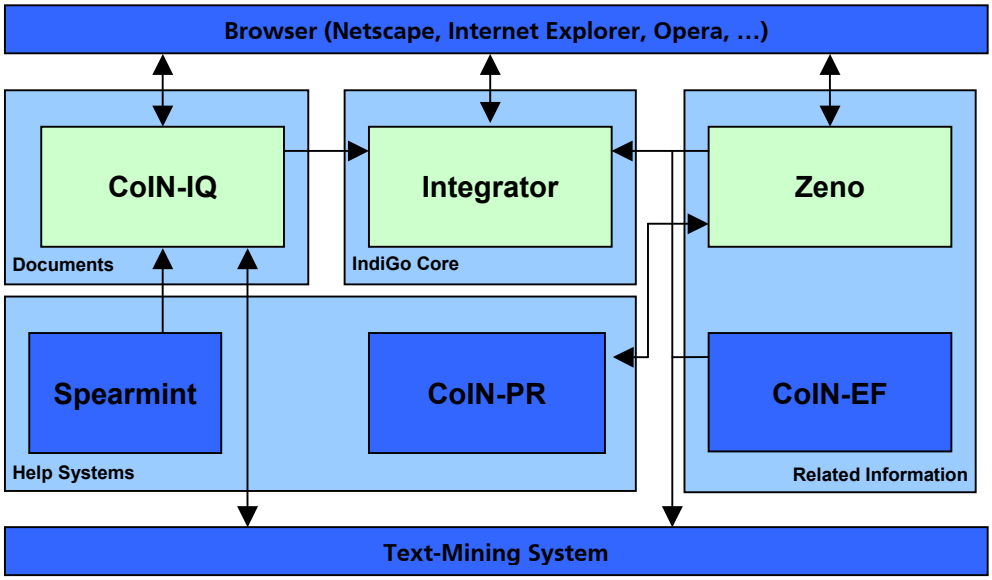

Figure 2: Information flow in the indiGo platform (upper level presented at CeBIT 2002)

To enhance the functionality of indiGo we connected Zeno with CoIN-PR (CoIN Project Registry), a project repository that stores all information about the projects and associated users. Information about the projects include, for example, the project type (e.g., research & development, transfer or consulting), status, funding, project staff, project manager or the list of participating partners.

CoIN-PR delivers information about a specific user's current projects, which is used to index contributions in Zeno with a project context and to construct queries for CoIN-EF. Beside commenting the business process models, the user will have the opportunity to recall context-specific lessons learned from CoIN-EF. To support and enhance the various roles in indiGo text-mining tools will be applied to analyze the discussions in order to detect new, previously unknown or hidden information for moderators and other roles, especially with the goal to extend or improve the lessons learned and the process models.

Based on standard internet technology indiGo is a truly distributed system. While Zeno is hosted on a web server at Fraunhofer AiS in Sankt Augustin, Germany, the CoIN system family is located at and maintained by Fraunhofer IESE in Kaiserslautern, Germany.

4.1 The indiGo integrator

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 a document and related informa-

tion. Based on Perl the integrator is a CGI script that offers three fundamental functions that are called either by CoIN-IQ or Zeno:

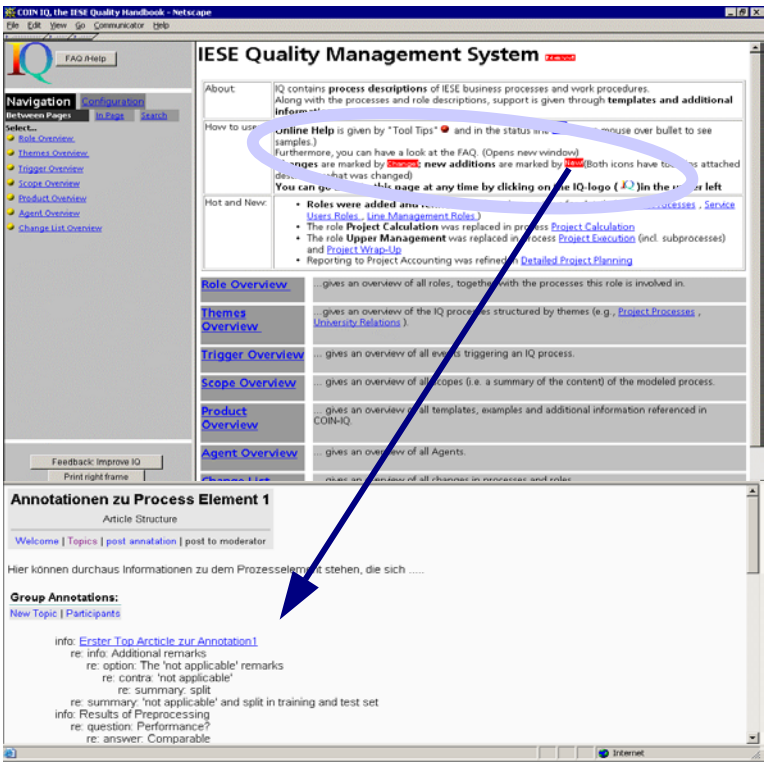

Figure 3: Split view with CoIN-IQ at the top and a related discussion in Zeno beneath

- Discuss: This function creates a split view upon a document and related information. In the current indiGo context this is a view on the specific business process model from CoIN-IQ in the upper part and beneath the appropriate discussion from Zeno.
- Annotate: Analogous to the previous function, the integrator creates a split view upon a business process model and a personal annotation for the current user.
- Destroy: To work with only one system this function collapses the split view of indiGo to a single frame. This is particularly helpful if the user wants to turn off the discussions from Zeno or if he switches into another discourse in Zeno that is not related to business processes.

4.2 CoIN-IQ

CoIN-IQ is IESE's business process model repository. 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).

A process within IQ is structured into "actions and subprocesses", "when to apply?", "objectives, results, and quality measures", "roles involved", "templates", "checklists", and "guidelines".

Overviews 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. 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 CoIN-IQ. Furthermore, changes or new objects in CoIN-IQ are marked by a "new" or "changed" icon (see top of figure 4). Finally, the most recent changes are announced on the entry page of CoIN-IQ.

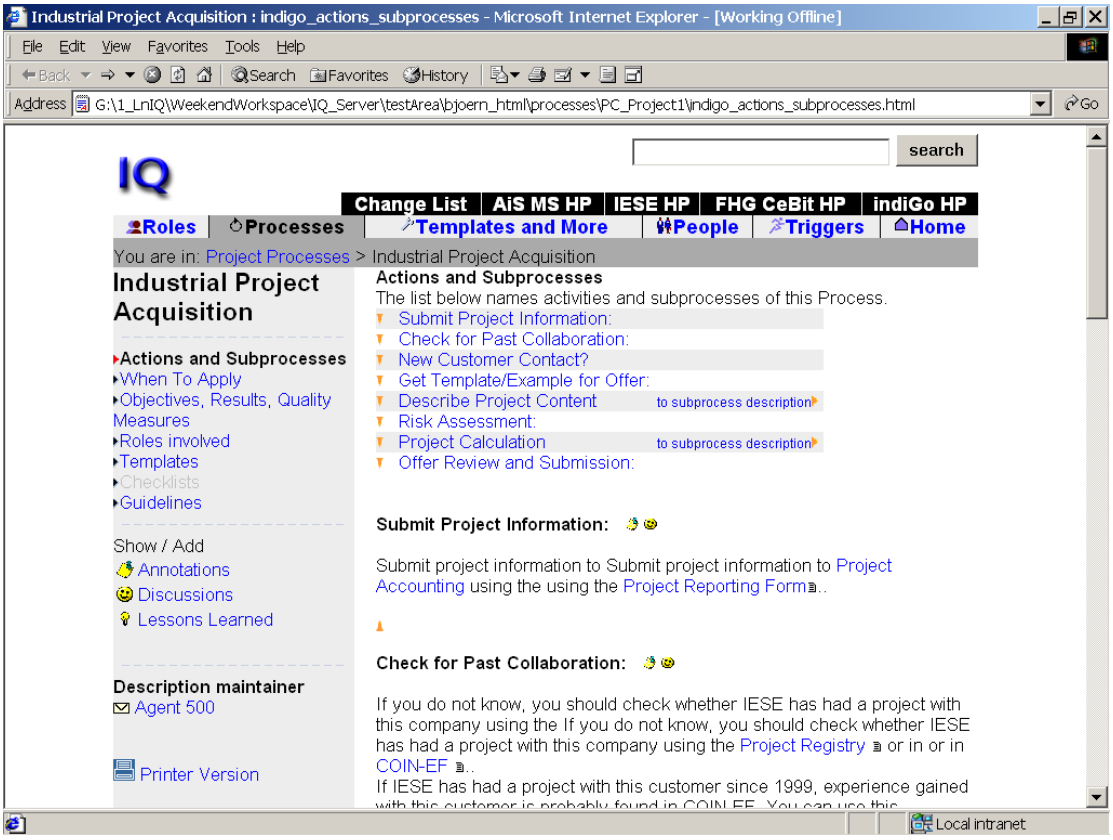

Figure 4: Screenshot of a process description. (Figure shows anonymized demonstrator)

4.2.1 CoIN-IQ for indiGo

In the indiGo platform, CoIN-IQ's start page is automatically generated by Zeno from articles in a special section for announcements. Other modifications of CoIN-IQ for indiGo concern the insertion of buttons for private annotations, group discussions and lessons learned. The buttons are displayed or hidden at the user's discretion. Buttons are inserted for entire processes and for all process elements. Internally, each process and element is identified by a unique number for the indiGo integrator and the other components; these number will not change even if the process model is reorganized.

4.2.2 Process model editors and publishing software

Spearmint is IESE's process modeling environment (Becker-Kornstaedt, Haman et al. 1999).

A Spearmint process model can be published on the web as an electronic process guide (EPG) with the process guidance tool EPG (Kellner, Becker-Kornstaedt et al. 1998). 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. CoIN-IQ is an instance of such an EPG.

In the following, based on Dellen, Könnecker, and Scott (2000), relevant process modeling editors and publication software are summarized. From the perspective of process learning, software 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 to annotate or discuss process models, and

(c) software that focuses on the collaborative creation of process models, i.e. 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 a two-way communication between process engineer or author and organizational members. (c) concentrates on supporting process engineers and authors in the creation of process models, which in practice will also include discussions.

For each of those categories, table 1 gives some examples. Process Model (No 1) belongs to category (a). It is focused in 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) (Scott, Jeffery & Becker-Kornstaedt 2001) (No 4). DaimlerChrysler's LID-system (von Hunnius 2000) allows public annotation of software process models, which the process engineer can distill to lessons learned and attached to the process model (No 5). Finally, as representatives of category (c) ARIS and ADONIS focus on collaborative editing and publishing of graphical represented business process models. ARIS also offers support for enacting the business process models, e.g. via Lotus Notes.

Name	No	**Publi-cation**	**Anno-tation**	**Discu**	**Coll. Crea-tion**	**URL / further information**
Process Model	1	X				www.processmodel.com
Process	2	X				www.scitor.com/pv3/purchase.process.asp
SPEARMINT / Annotation	3	X	X	X		www.iese.fhg.de/Spearmint_EPG/
SPEARMINT /	4	X	X	X		www.iese.fhg.de/Spearmint_EPG/

PageSeeder						
LID System	5	X	X	X		(von Hunnius (2000)
ProcessWiseIntegrator	6	X			X	[report.ps]
ADONIS	7	X	?	?	X	www.boc-eu.com
INCOME	8	X			X	www.promatis.de
INNOVATOR	9	X			X	www.mid.de
in-Step	10	X			X	www.microtool.de
Aris Web Designer	11	X	X?	X?	X	www.ids-scheer.de

Table 1: Overview of process modeling and publication software

4.3 Zeno

Turning from tools for process models to tools for discussion, the objectives and major concepts of Zeno can be motivated.

4.3.1 Software for document-centered discourses on the web

Zeno is an e-participation platform (www.e-partizipation.org) (Voss 2002) with a spectrum of functions that comprises and extends

(a) simple threaded discussions

(b) document-centered discourses

(c) information structuring during group decision making

Most electronic discussion forums, like the ones mentioned above but also newsgroups, support simple threaded discussions (a). Some tools, e.g. http://icommons.harvard.edu/, recognize URLs or even HTML tags in the contributions or allow to attach documents.

D^3E belongs to category (b). It can process any hierarchical HTML file into a frames-based environment with automatic hyperlinking for navigating around sections, checking citations and footnotes, and tight integration with a discussion space for critiquing documents. Moderators may influence the look and feel of a discussion space, they may edit, hide, or delete contributions. D3E is available as open source (http://d3e.sourceforge.net/) (Sumner & Buckingham Shum 1998). The e-learning platforms Hyperwave eLearning SUITE supports annotations and discussions of course units. Moreover, it offers a set of labels to characterize contributions as notes, questions, responses, acceptance and rejection (www.hyperwave.com).

Predefined labels for qualifying contributions are more familiar in tools for group decision making (c), especially for brainstorming (www.facilitate.com). Softbicycle's QuestMap (www.softbicycle.com) distinguishes questions, ideas, pros, cons, decisions, notes, and ref-

erences, a variant of the famous IBIS grammar (Kunz & Rittel 1970) which was first implemented in gIBIS (Conklin & Begemann 1988). Tools in this category usually allow to restructure the contributions, i.e. they support maps rather than threads, deliberative argumentation rather than spontaneous reaction.

The first version of Zeno, which also supported a variant of IBIS (Gordon & Karakapilidis 1999), was presented at CeBIT 1996 and continuously improved up to version 1.9 in 1999. Since then a completely new system has been realized addresses a broader spectrum of discourses in the knowledge society: participatory problem solving, consensus building (Voss, Röder & Wacker, 2002), mediated conflict resolution (Märker, O., Hagedorn, H., Trénel, M. & Gordon, 2002), teaching and consulting. The new Zeno focuses on e-discourses and supports e-moderators 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 (Erickson 1999). (Turoff et al. 1999) 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.

4.3.2 Zeno concepts

As a consequence, Zeno distinguishes three kinds of objects: sections to tailor the settings for an e-discourse, articles as units of a communication (contributions), and links as directed relations between articles or even sections.

Moderators specify the readers, authors, and co-editors of the section, its discourse grammar, a style sheet to control the presentation, and plugged-in functionality (for mapping, awareness, polling, etc).

An article has a title, usually a note (plain text or html), and possibly document attachments. From its author it may get a label to indicate its pragmatic (or ontological) role in the discourse (e.g. issue, option, criterion, argument, decision, summary, question, comment), and it may receive an additional qualifier from the moderator (e.g. green, yellow, red cards). Articles may be selected (and deselected) as topics and may be ranked to influence their ordering. An article may have temporal references (to be displayed on a timeline), keywords (to be searched together with the title and note), and attributes related to its visibility and accessibility.

Links between articles or sections may be labeled to express relations, such as refers-to, responds-to, justifies, questions, generalizes, suggests, pro, contra) so that complex networks (or hyperthreads) can be built. Links between Zeno articles and sections are visible at both end points and can be traversed in both directions. They are automatically maintained by Zeno, so moderators may edit, copy and move groups of articles with their links.

Zeno links may also point to external web resources; they are used for document references in indigo and for spatial references (to be displayed on a map) in KogiPlan

(www.kogiplan.de).

Users are received on a personal home page. Here they can bookmark and subscribe sections in order to be notified of their latest contributions. Each section offers different views: the latest articles, the topics, the complete article structure, a sorted list of articles as a result of a full-text search, the hierarchy of subsections, or the timeline. Authors may create or respond to articles in a section, and moderators may edit, move and copy articles, change links and assign labels, and manipulate sections. Users and groups are administered through an address book.

Zeno can be assessed from any regular web browser without any local installations. The Zeno server is implemented on top of open source products: tomcat as web server and servlet runner, velocity for templates in the user interface, Java for the kernel, and MySQL for the data base. Zeno itself is available as open source (http://zeno.berlios.de/).

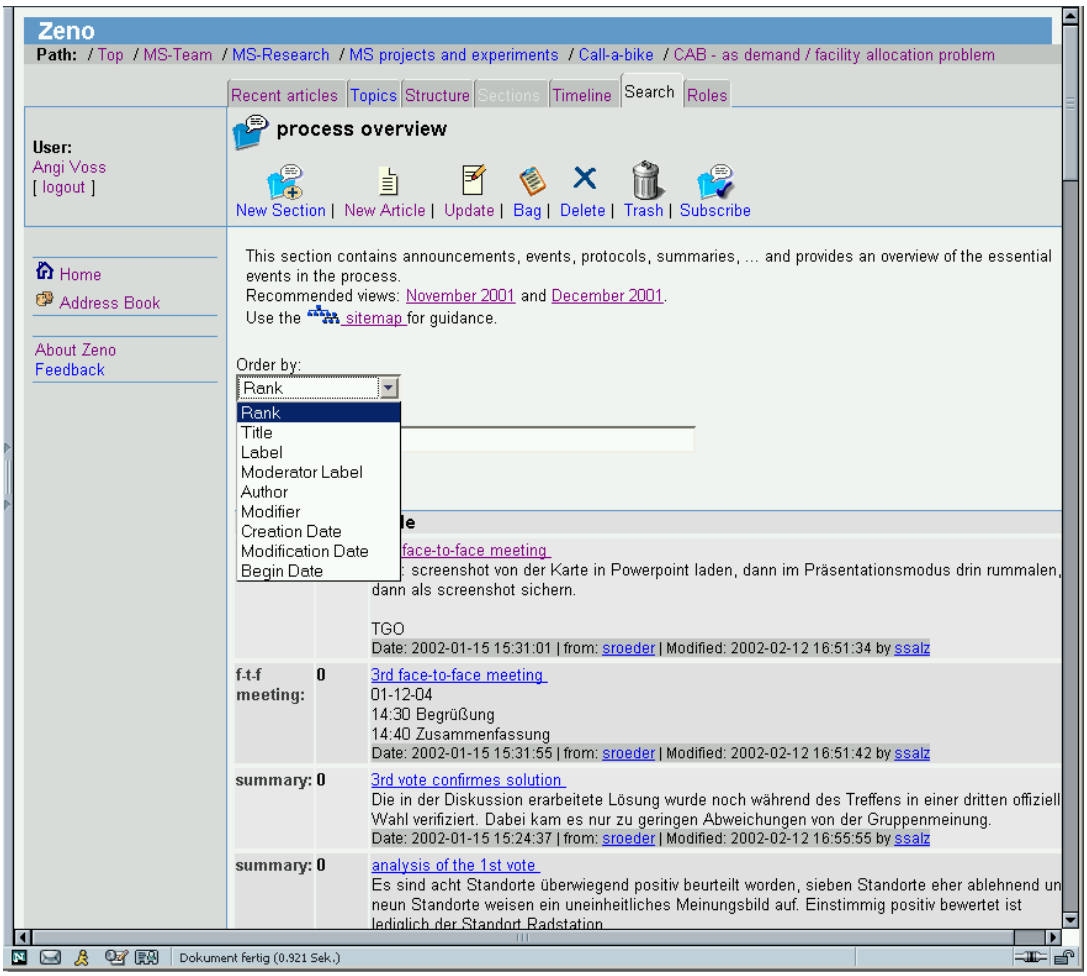

Figure 5: The search view in the overview section of a spatial decision making discourse in Zeno

4.3.3 Zeno for indiGo

In Zeno, document-centered discourses, or more specifically, discourses about process models, are made possible through the indiGo integrator and some indigo-specific adaptations of Zeno.

The structure and ordering of process models and their elements is reflected in the hierarchies of sections and their ranking. The mapping between these structures is accomplished through Zeno links, the names of which encode identifiers for the process model and element.

Moderators first create entries for users and groups in the address book. Next, to generate a section for discussing a process, the moderators click on the “discussion” button of the process or any of its elements and then select a group as readers and writers for the discussion. Subsections for discussing process elements are created on demand, when users click on the associated processes and selects the discussion group. The subsections inherit the discourse grammar of their super-section and are restricted to the selected group as authors.

When a user clicks on an “annotation” button for the first time, a personal section is created. This section and its subsections can only be accessed by this user with all rights of a moderator. Subsections for processes and their elements are again created on demand, when the user clicks on the corresponding “annotation” buttons.

The start page of the indiGo system is automatically generated. The upper part displays announcements. These are articles in a section called “StartPage” , can be edited by all indiGo moderators. Beneath the announcements, the start page lists all new articles in the user’s discussion groups. This service replaces the subscription and notification mechanism that is otherwise available on the users’ personal home page in Zeno.

observation, a problem, guideline, pragmatic solution, or an improvement suggestion

For the introduction and operational phases different discourse grammars will be available. “info”, “question”, “comment”, “suggestion”, “example” are the article labels during introduction, “observation”, “problem”, “suggestion”, “solution”, “example” and “summary” are the article labels during operation. Link labels are in both phases “re”, “pro”, “con”, “see also”. Qualifier will include “closed” to indicate threads with a conclusion, and “invalid” to indicate threads that may have become invalid due to modifications of the process model. To come back to the introductory example, Ms Legrelle could have attached a “problem” to the guideline on payment schedules, “re”sponded with a “suggestion” concerning small start-ups, and supported it with a ”pro” “example” from the Orion project.

4.4 CoIN-EF

Compared to the objectives of an organization as captured in its process models, projects have a short-term perspective, oriented towards the goals of the project. Therefore an organizational unit that is responsible for experience management is required and has to be separated from the project teams. As already mentioned, such a separate organizational unit is called experience factory (EF), which for the IESE is operationalized by the CoIN team. Within the integrated experience base (EB), all kinds of experience necessary for daily business are stored (e.g., guidelines, or observations). Defined processes populate the EB systematically with experience typically needed by IESE’s project teams. The retrieval of experiences from the EB is planned right at the start of the build-up and supports a goal-oriented, context-sensitive, similarity-based retrieval of different kinds of interrelated experiences.

Within CoIN-EF, lessons learned (LL) about project management are captured. An LL can take on the form of an observation, a problem, guideline, pragmatic solution, or an improve-

ment suggestion. Each LL is personalized to allow a querying IESE member to ask a colleague for further information. The context of these LLs is modeled by the two concepts “project” and “process”. A “project” is a characterization of the project where the lesson learned was gained (e.g., person months, duration). The “process” names the business process and thus the project phase in which the LL was gained. Therefore, project team members can specify their current environment as well as the current situation to search the EB for similar experiences.

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

The technical infrastructure, called INTERESTS (INTElligent REtrieval and STorage System), is shown in figure 6. It consists of a tool layer for accessing and presenting the EB contents using a standard web browser, a general purpose EB server, and a commercial CBR tool (orenge from empolis, Germany), which is used for the actual EB.

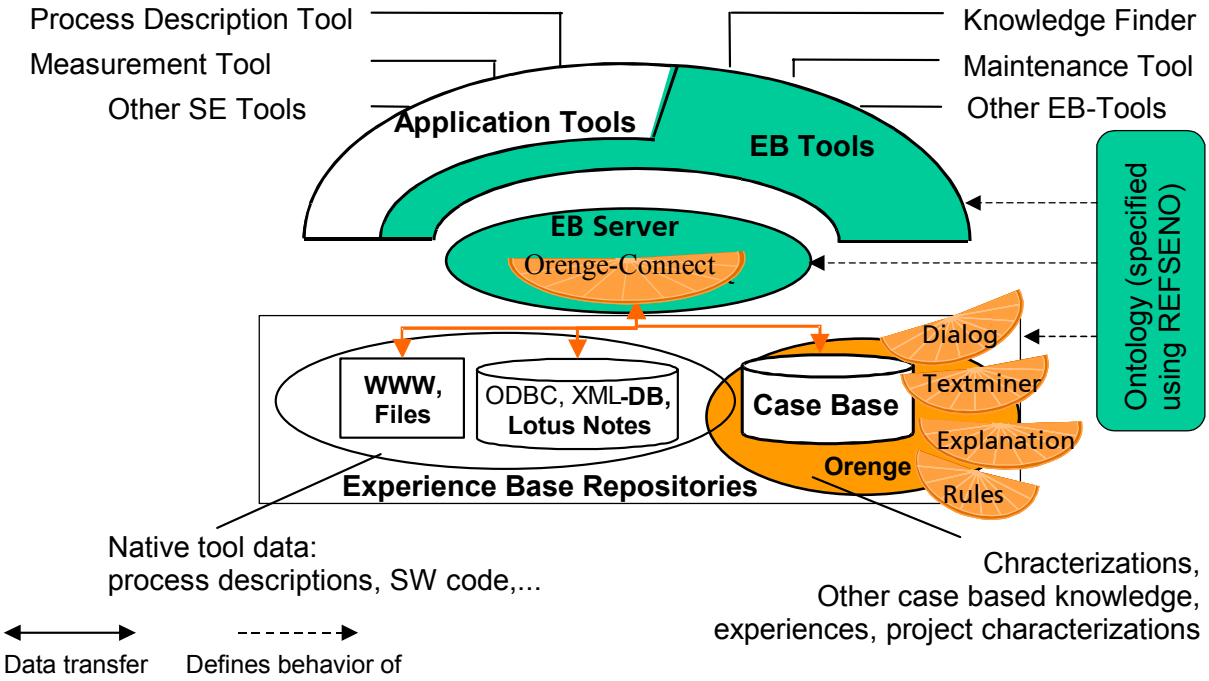

Figure 6: CoIN’s technical infrastructure (INTERESTS)

4.4.1 CoIN-EF for indigo

CoIN-EF has not yet been integrated into the indigo platform. When a user presses an “experiences” button in CoIN-IQ, a query must be invoked via the integrator to CoIN-EF, which contains the respective process element and the user’s context. More challenging will be the integration with tools for knowledge construction: from discourses to experiences and from experiences to process models. As a preparation, the discourse grammar for the operational phase has been designed according to the formats for lesions learned. This should facilitate a mapping of the articles in a discussion to types of lessons learned.

4.4.2 Case-based reasoning for sharing process and project knowledge

Since several years there has been a strong tendency in the case-based reasoning (CBR) community (Kolodner 1993) to develop methods for dealing with more complex applications. One example is the use of CBR within knowledge management (KM) (Aha, Becerra-Fernandez et al. 1999). Another one is the integration of CBR with experience factories (Henninger 1995, Althoff & Wilke 1997, Tautz & Althoff 1997, Bergmann, Breen et al. 1999). The latter also contributed to the development of the experience management subfield of KM (Tautz 2000, Bergmann 2001, Althoff, Decker et al. 2001), which already found one implementation through the merger of the German CBR and KM communities (www.experience-management.org, Minor & Staab 2002). Meanwhile many papers have been published that are related to the use of CBR in KM. (Weber, Aha, and Becerra-Fernandez 2001) give an overview on intelligent lessons learned systems, which includes CBR approaches. While (Wargitsch 1998) describes how CBR can be used for workflow support, (Chen-Burger, Robertson, and Stader 2000) focus on the support for business modeling in general. (Decker and Jedlitschka 2001) present a first step how business processes and EM/CBR can be integrated. Further approaches on process-oriented knowledge management and CBR can be found in (Weber and Gresse von Wangenheim 2001). CBR-based knowledge reuse for project management is described in (Althoff, Nick, and Tautz 1999), (Tautz 2000), (Brandt and Nick 2001), and (Friedrich, Iglezakis et al. 2002). CBR for supporting knowledge mediation is the topic underlying (Griffiths, Harrison, and Dearden 1999).

4.5 Text Mining in indiGo

Text mining is concerned with the task of extracting relevant information from natural language text and to search 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 (Hřebíčček 1996). 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 data 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 (Joachims 1998). AIS has successfully applied SVM to different classification problems - topic detection and author identification (Kindermann, Diederich et a. 2002), multi-class classification (Kinder-

mann, Paaß & Leopold 2001) - on different linguistic corpora: Reuters newswire, English and German newspapers (Leopold & Kindermann 2002), as well as radio-broadcastings (Eickeler, Kindermann et al. 2002). The major problem of applying text classification techniques in the indiGo project is the amount of data. The training of a SVM requires some hundred positive and negative examples for each class to be considered. These data must be collected in the group discussions. The contributions in a discussion group have to annotated with respect to the desired classes by the moderator.

An especially challenging task to text mining systems is to map the unstructured natural text to a structured internal representation (basically a set of data objects). indiGo requires to map text documents generated in the group discussions to structured information of project experiences. However, the limited scope of the indiGo-project - many roles can only be fulfilled by a finite number of subjects (e.g. the number of IESE's employees or costumers is finite) - makes it possible to invent simplifying solutions to many problems, which are not feasible in the general case.

The context of an utterance consists of all elements in a communicative situation that determine the understanding of an utterance in a systematic way. Context divides up into verbal and non-verbal context (Bußmann 1990). Non-verbal context cannot - or at best to a small extent - be conveyed in written text. Abstracting away from the non-verbal context of the situation which a text (spoken or written) is produced, means, that the lost information has to be substituted by linguistic means in order to avoid misunderstandings resulting from the loss of information. This is why spoken and written language differ. Speaker and hearer are exposed to the same contextual situation, which disambiguates their utterances, whereas writer and reader - in the traditional sense of the word - are not.

Computer-mediated communication adopts an intermediate position in this respect. Writer and reader react on each other's utterances as speaker and hearer do. They are in the same communicative situation. But their opportunity to convey non-verbal information is limited as well as the chance to obtain information about the contextual situations of their counterparts.

The context of the communicative situation becomes crucial in the IndiGo setting when discussions are condensed to project experiences. The communicative situation of the discussion is lost and respective information has to be added to the natural language data. This limits the degree of information compaction of linguistic data. Consequently the decontextualization suggested in figure 1 has to be carefully performed in order to not end up in compressed but nevertheless senseless "structured information". How and to what extent information about the communicative situation can be concentrated or discarded is an interesting research objective of the indiGo project.

To provide the moderator with information about the problem-orientation of the participants in a discussion we propose an “index of speciality of language”, which can be calculated on the basis of the agreement of the vocabulary of writer and reader. Self-organizing maps (SOM) (Kohonen 2001) (Merkl 1997) can give an overview over a set of documents, and thus inform the moderator about similar themes that are discussed in different threads. Standard clustering procedures as well the hierarchical analysis of textual similarities (Mehler 2002) can enhance the presentation of textual data in order to support the moderator in formalizing discussion contributions as reusable experiences or cases.

5. Outlook

indiGo was designed to support all kinds of knowledge that have been identified as being import for process learning, namely process models (with their associated templates), experiences from instantiating process models in concrete projects, discussions about processes in closed or open groups, and private annotations of process models. Thus with indiGo, any concerned organization member can make private annotations for a newly introduced, or changed, business process model. Staff can decide which of the issues that attracted their attention should be discussed within a selected group of people. The indiGo technical infrastructure enables the organization of various of such discussion groups based on a customizable discourse grammar, and indiGo's e-moderation method guarantees that such discussions are carried in a structured and goal-oriented manner. This helps to identify valuable experiences, which then are represented as semi-formal cases, and stored in the experience base. Using case-based reasoning, these experiences are then available for both process improvement/change and process execution.

The first version of indiGo was presented in March 2002 at CeBIT. Starting in April 2002, indiGo will be validated within a case study carried out at Fraunhofer IESE in Kaiserslautern, Germany. New project and strategy processes will be introduced for the whole institute and indiGo has been chosen as the process learning platform. We expect very valuable feedback for all the described indiGo methods and technologies.

In parallel, specified but not yet implemented features will be realized. For instance, if a process model is modified or reorganized, the corresponding annotations and discussions should automatically be marked for re-validation or be reorganized accordingly. In parallel, the indiGo platform will be extended to include the components on the lower level in figure 4, starting with CoIN-EF.

As soon as discussions will become available from the case study, text mining experiments can begin. For that purpose, the discussions in Zeno will be exported in GXL, an XML dialect for graph structures. Private annotations remain private and will not be subject to text mining.

Beyond the current project we consider the possibility to extend the indiGo approach to applications where process models do not play such a central ("backbone") role. Although a platform for organizational learning should eventually cover all knowledge categories treated in indiGo, the first steps to organizational learning need not necessarily involve process models. Maybe, an organization would first like to invest into an experience base or into a communication platform, and add process models only later. The challenging research question here is, to which degree indiGo's methods and technologies can still be applied or easily tailored to such an organization's needs.

6. References

Aha, D.W., Becerra-Fernandez, I., Maurer, F., and Muñoz-Avila, H. (Eds.) (1999). Exploring Synergies of Knowledge Management and Case-Based Reasoning: Papers from the AAAI 1999 Workshop (Technical Report WS-99-10). Menlo Park, CA: AAAI Press.

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.

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. Juli 2001, Institut für Bildverarbeitung und angewandte Informatik

Althoff, K.-D., Feldmann, R. & Müller, W. (eds.) (2001). *Advances in Learning Software Organizations.* Springer Verlag, LNCS 2176, September 2001.

Althoff, K.-D., Nick, M. & Tautz, C. (1999). An Application Implementing Reuse Concepts of the Experience Factory for the Transfer of CBR System Know-How. In: E. Melis (ed.), *Proc. of the Seventh German Workshop on Case-Based Reasoning (GWCBR'99).*

Althoff, K.-D. & Wilke, W. (1997). Potential Uses of Case-Based Reasoning in Experience Based Construction of Software Systems and Business Process Support. In: R. Bergmann & W. Wilke (eds.), *Proc. of the 5th German Workshop on Case-Based Reasoning,* LSA-97-01E, Centre for Learning Systems and Applications, University of Kaiserslautern, 31-38.

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.

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.

Ulrike Becker-Kornstaedt. Towards Systematic Knowledge Elicitation for Descriptive Software Process Modeling. In Frank Bomarius and Seija Komi-Sirviö, editors, Proceedings of the Third International Conference on Product-Focused Software Processes Improvement (PROFES), Lecture Notes in Computer Science 2188, pages 312-325, Kaiserslautern, September 2001. Springer.

U. Becker-Kornstaedt and W. Belau. Descriptive Process Modeling in an Industrial Environment: Experience and Guidelines. In R. Conradi, editor, *Proceedings of the 7th European Workshop on Software Process Technology* (*EWSPT 7)*, Kaprun, Austria, pages 177–189, Lecture Notes in Computer Sciences, Springer-Verlag. 2000.

Bergmann, R. (2001). Experience management - foundations, development methodology, and internet-based applications. Postdoctoral thesis, Department of Computer Science, University of Kaiserslautern (submitted).

Bergmann, R., Breen, S., Göker, M., Manago, M. & Wess, S. (1999). *Developing Industrial CBR Applications. The INRECA-Methodology*. Springer Verlag, LNAI 1612.

Brandt, M. & Nick, M. (2001). Computer-Supported Reuse of Project Management Experience with an Experience Base. In *Althoff, Feldemann & Müller (2001)*, 178-191.

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

Conklin, J. and M. Begeman (1988) “gIBIS: A Hypertext Tool for Exploratory Policy Discussion", *Transactions of Office Information Systems*, 6(4): 303-331.

Decker, B. & Jedlitschka, A. (2001). The Integrated Corporate Information Network iCoIN: A Comprehensive Web-Based Experience Factory. In *Althoff, Feldmann & Müller (2001)*, 192-206.

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

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

Erickson, T. (1999): “Persistent Conversation. An introduction“. Journal of Computer-Mediated Communication, 4.

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.

Friedrich, R., Iglezakis, I., Klein, W. & Pregizer, S. (2002). Experience-based decision support for project management with case-based reasoning.. In Minor & Staab (2002), 139-150.

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.*

Griffiths, A.D., Harrison, M.D. & Dearden, A.M. (1999). Case Based Reasoning Systems for Knowledge Mediation. in M. A. Sasse and C. Johnson (eds.), *Proceedings of Interact 99*, (Edinburgh), IOS Press. pp. 425-433.

Henninger, S. (1995). Developing Domain Knowledge Through the Reuse of Project Experiences. *Proc. Symposium on Software Reusability (SSR '95), Seattle WA, April 1995.*

Hřebíčček, Luděk & Altmann, Gabriel(1996): The Levels of Order in Language, Glottometrika 15, 38-61.

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

Marc I. Kellner, Ulrike Becker-Kornstaedt, William E. Riddle, Jennifer Tomal, and Martin Verlage (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.

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.

Kindermann, Jörg & Diederich, Joachim & Leopold, Edda & Paaß, Gerhard: *Identifying the Author of a Text with Support Vector Machines*; accepted at *Applied Intelligence*.

Kohonen, Teuvo (2001): Self-organizing maps, Springer: Berlin, Heidelberg, New York

Kolodner, J. L. (1993). *Case-Based Reasoning*. Morgan Kaufmann.

Kunz, Werner und Horst W.J. Rittel (1970): "Issues as elements of information systems." Center for Planning and Development Research, Institute of Urban and Regional Development Research. Working Paper 131, University of California, Berkeley.

Leopold, Edda & Kindermann, Jörg (2002): *Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?;* in: *Machine Learning* 46, 423 - 444.

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.

Mehler, Alexander (2002): Hierarchical analysis of text similarity data, in: Thorsten Joachims & Edda Leopold (eds.): Künstliche Intelligenz, Schwerpunkt Textmining, to appear.

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).

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).

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, pp55-58.

Sumner, T. & Buckingham Shum, S. (1998) “From Documents to Discourse: Shifting Conceptions of Scholarly Publishing“, Proc. CHI 98: Human Factors in Computing Systems, 18-23 April, 1998, Los Angeles, CA. ACM Press: New York

Tautz, C. (2000). *Customizing Software Engineering Experience Management Systems to Organizational Needs*. Doctoral Dissertation, Department of Computer Science, University of Kaiserslautern, Germany.

Tautz, C. & Althoff, K.-D. (1997). Using Case-Based Reasoning for Reusing Software Knowledge. *Case-Based Reasoning Research and Development – Second International Conference (ICCBR97),* Providence, RI, Springer Verlag.

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.

von Hunnius, J.-P. (2000). WESPI – Web Supported Process Improvement. In K.-D. Althoff & W. Müller (eds.), Proc. 2nd Internationl Workshop on Learning Software Organizations, June 20, 2000, Oulu Finland, Fraunhofer IESE, Kaiserslautern, Germany.

Voss, A. (to appear (2002)): „Zeno – Software für Online-Diskurse in der Mediation”. In Online-Mediation. Theorie und Praxis computer-unterstützter Konfliktmittlung(Eds, Märker, O. and Trenél, M.) Sigma, Berlin.

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.

Wargitsch, C. (1998). *Ein Beitrag zur Integration von Workflow- und Wissensmanagement unter besonderer Berücksichtigung komplexer Geschäftsprozesse*. Doctoral Dissertation, University of Erlangen-Nürnberg, Germany.

Weber, R., Aha, D.W., & Becerra-Fernandez, I. (2001). Intelligent lessons learned systems. Expert Systems with Applications, 20, 17-34.

Weber, R. & Gresse von Wangenheim, C. (eds.) (2001). Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning, Technical Note AIC-01-003, Washington, DC: Naval Reasearch Laboratory, Naval Research Center for Applied Research in Artificial Intelligence.

Yun-Heh Chen-Burger, Dave Robertson & Jussi Stader (2000). A Case-Based Reasoning Framework for Enterprise Model Building, Sharing and Reusing. *Proc. ECAI 2000 Workshop on Knowledge Management and Organizational Memories*, Berlin.