KNOWLEDGE MANAGEMENT

AND

ORGANIZATIONAL MEMORIES

Workshop Co-Chairs:

Rose DIENG-KUNTZ

INRIA, Sophia-Antipolis

and

Nada MATTA

Université de Technologie de Troyes

Preface

Knowledge Management (KM) is one of the key progress factors in organizations. It involves explicit and persistent representation of knowledge of (geographically) dispersed groups of people in the organization, so as to improve the activities of the organization. Although KM is an issue in human resource management and enterprise organization beyond any specific technology questions, there are important aspects that can be supported or even enabled by intelligent information systems. Especially AI and related fields provide solutions for important parts of the overall KM problem.

Identification and analysis of a company's knowledge-intensive work processes (e.g., product design or strategic planning). Knowledge Engineering and Enterprise Modeling techniques can contribute to this topic. The analysis of information flow and involved knowledge sources allows to identify shortcomings of business processes, and to specify requirements on potential IT support.

In an organization, know-how may relate to problem solving expertise in functional disciplines, experiences of human resources, and project experiences in terms of project management issues, design technical issues and lessons learned. The coherent integration of this dispersed know-how in a corporation, aimed at enhancing its access and reuse, is called "corporate memory" or "organizational memory" (OM). It is regarded as the central prerequisite for IT support of Knowledge Management and is the means for knowledge conservation, distribution, and reuse. An OM enables organizationallearning and continuous process improvement.

Activities underlying knowledge management in an organization can comprise detection of needs, construction, distribution, use and maintenance of the corporate memory. It demands abilities to manage disparate know-how and heterogeneous viewpoints, to make it accessible and suitable for adequate members of the organization. When the organization knowledge is distributed onseveral experts and documents in different locations all over the world, the Internet or an Intranet inside the organization and World Wide Web (WWW) techniques can be a privileged means for acquisition, modelling, management of this distributed knowledge.

Examples of interesting topics for organizational memories are:

- Dimensions of knowledge management: organization, competence, methodology...
- Enterprise modeling
- Artificial Intelligence methods or techniques for construction of computational corporate memories (knowledge bases, case bases, intelligent documentary systems, agent-based systems...)
- Business Intelligence Solutions for KM
- Intranet Solutions for KM
- Document Management Solutions for KM
- MultiMedia solutions for KM
- Content Management solutions for KM
- Architectures for KM/OM systems
- Integration of formal and informal knowledge in KM/OM
- Integration of knowledge from different groups in an organization
- Knowledge sharing between different groups in an organization (possibly via Internet/Intranet)

- Cooperative (possibly Web-based) building, adaptation and evolution of a corporate memory
- Building and Exploiting a Corporate Semantic Web
- Web-based repositories for sharable ontologies and reusable problem-solving methods
- Web-based terminology servers
- Assessment of concrete applications for knowledge management
- Case studies of building KM/OM in enterprises
- Active, context-dependent knowledge supply

Workshop organizing committee

Jean-Paul Barthès
Université de Technologie de Compiègne
BP 20529, F-60205 Compiègne
Tel +33 3 44 23 44 66, Fax +33 3 44 23 46 33
E-mail: Jean-Paul Barthes <barthes@utc.fr>

Rose Dieng-Kuntz (co-chair)
INRIA Acacia Project, 2004 route des Lucioles, BP 93
06902 Sophia-Antipolis Cedex FRANCE
Phone: +33 4 92 38 78 10 , Fax: +33 4 92 38 77 83
E-mail: Rose.Dieng@sophia.inria.fr

Knut Hinkelmann
University of Applied Sciences Solothurn
Riggenbachstrasse 16
CH-4600 Olten
Phone: +41-(0)62-286 00 80 Fax: +41-(0)62-296 65 01
Email: knut.hinkelmann@fhso.ch

Ann Macintosh
International Teledemocracy Centre
Napier University
10 Colinton Road
Edinburgh, EH10 5DT
Tel.: +44 (0) 131 455 2421 Fax: +44 (0) 131 455 2282
Email: A.Macintosh@napier.ac.uk

Nada Matta (co-chair)
Université de Technologie de Troyes (GSID/Tech-CICO)
12, rue Marie Curie BP. 2060,
10010 Troyes Cedex France
Tel: (+33) 3 25 71 58 65
E-mail:nada.matta@univ-troyes.fr

Ulrich Reimer
Swiss Life Information Systems Research Group Postfach CH-8022
Zurich, Switzerland
Phone: +41 1 7114061 Fax: +41 1 7116913
E-mail: Ulrich.Reimer@swisslife.ch

Carla Simone
Department of Informatics, Systems and Communication
University of Milano-Bicocca(Italy)
Phone: +39 011 6706736 Fax: +39 011 751603
E-mail: simone@di.unito.it

Program committee

Mark Ackerman, University of California, Irvine (USA)
Hans Akkermans, Free University of Amsterdam, The Netherlands
Richard Benjamins, Isoco, Spain
Joost Breuker, University of Amsterdam, the Netherlands
John Debenham, University of Technology, (Sydney, Australia)
John Domingue, Open University, (UK)
Jean-Louis Ermine, Tech-CICO, UTT (Paris, France)
Jérôme Euzenat, INRIA Rhône-Alpes (Grenoble, France)
Fabien Gandon, INRIA-Sophia-Antipolis (France)
Tom Gruber, Intraspect (USA)
Robert Jasper (USA)
Gilles Kassel, LARIA, université d'Amiens (France)
Myriam Lewkowicz, Tech-CICO, UTT (Troyes, France)
Dirk Mahling, University of Pittsburgh (USA)
Frank Maurer, University of Calgary (Canada)
Myriam Ribière (Motorola, France)
David G. Schwartz, Bar-Ilan University (Israel)
Nigel Shadboldt, University of Southampton, UK
Steffen Staab, University of Karlsruhe, (Germany)
Rudi Studer, University of Karlsruhe, (Germany)
Gertjan van Heijst, CIBIT (Utrecht, The Netherlands)
Manuel Zacklad, Tech-CICO, UTT (Troyes, France)

Acknowledgements

We thank very much the members of the organization committee and of the program committee for their careful reviews of the 21 papers submitted. We also thank very much Sophie Honnorat (INRIA, Sophia-Antipolis) for her very efficient logistic support for the preparation of this manuscript and for the maintenance of the Web site of the workshops.

TABLE OF CONTENTS

PROGRAM
Monday, July 22nd

8:45 Introduction (Rose Dieng-Kuntz, Nada Matta)

<u>SESSION I:</u> Enterprise modelling and Organizational Learning

9:00 to 9h30
An Intellectual Genealogy Graph : Affording a Fine Prospect of Organizational Learning
Mitsuru Ikeda, Hiroyuki Tsumoto, Riichiro Mizoguchi

9:30 to 10:00
Enterprise Modelling: A Declarative Approach for FBPML
Yun-Heh Chen-Burger, Austin Tate, Dave Robertson

10:00 to 10:20
Morning refreshments

<u>SESSION II:</u> Ontologies and Documents

10:20 to 10:50
Initiating Organizational Memories using Ontology Network Analysis
Yannis Kalfoglou, Harith Alani, Kieron O'Hara, Nigel Shadbolt

10:50 to 11:20
To supply organization views, suited to users: an approach to the design of organizational memories
J-Y.Fortier, Cormier, G.Kassel, C.Barry, C.Irastorza, S.Bruaux

11:20 to 11:50
Enhancing Experience Management and Process Learning with Moderated Discourses: the indiGo approach
Klaus-Dieter Althoff , Ulrike Becker-Kornstaedt, Björn Decker, Andreas Klotz, Edda Leopold, Jörg Rech, Angi Voβ

11:50 to 12:20
Knowledge Management Performance Index Considering Knowledge Cycle
Kun Chang Lee, Namho Chung, Soochoul Joung, Byung-uk Kang

12:20 to 14:00
Lunch

<u>SESSION III:</u> Knowledge Models

14:00 to 14:30
The PROMOTE Approach: Modelling Knowledge Management Processes to describe organizational knowledge systems
Dimitris Karagiannis, Robert Woitsch

14:30 to 15:00
Continuous capitalisation of design knowledge
Nada Matta, Benoit Eynard, Lionel Roucoules, Marc Lemercier

15:00 to 15:30
Traceability and knowledge modelling
Smain Bekhti, Nada Matta

15:30 to 15:50
Afternoon refreshments

<u>SESSION IV:</u> Cooperative Approaches

15:50 to 16:20
A political model of co-operative production of knowledge in design process: the Shared Medical File (SMF)
Eddie Soulier, Corinne Grenier

16:20 to 16:50
A Cooperative Approach to Corporate Memory Modeling
Jesualdo Tomas Fernandez Breis, Rodrigo Martinez-Béjar, Laura Maria Campoy-Gomez, Fernando Martin Rubio

<u>SESSION V:</u> Databases

16:50 to 17:20
Inductive-Deductive Databases for Knowledge Management
Marcello Aragão, Alvaro Fernandes

<u>SESSION VI:</u> Applications

17:20 to 17:50
Developing an Intranet-based Knowledge Management Framework in a Consulting Firm: a Conceptual Model and its Implementation
Reena Sarkar, Somprakash Bandyopadhyay

17:50 to 18:30
Final discussions

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]

Abstract. The indiGo project aims at improving process knowledge by successive consolidation of feedback, ranging from private annotation, through structured communication in communities of practice, to improved process models and lessons learned. It develops a methodology and integrates previously independent software for process modeling, moderated discourses, experience management and text mining. Both will be evaluated in case studies.

1 INTRODUCTION

The business process models of organizations operating in innovative, knowledge-intensive or service-oriented markets are one of their major knowledge assets and a competitive advantage. 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.

2 THE FRAMEWORK

indiGo's key objective is to create and sustain living process models, that is, process models that are accepted by the organizations members, adapted to organizational changes on demand, and continuously enriched with experience from the operating business of the organization.

2.1 Example

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. A discourse is a deliberative, reasoned communication, it is focused and intended to culminate in group decision making (Erickson 1999). An e-discourse is text-based and conducted (partially) through internet technology. In e-discourses, more persons can participate, the audience may be distributed in space and time, vary in size, composition, background. However, in today's web-based discussion forums, a high tendency to incoherence, drift, and dissolution can be observed. To bridge this tradeoff between promise and reality additional value must be created, and this should be done by exploiting the persistent nature of e-discourses: they may be browsed, replayed, searched, annotated, visualized, analyzed, restructured, and recontextualized.

[1] Fraunhofer IESE, Sauerwiesen 6, D-67661 Kaiserslautern, althoff@iese.fraunhofer.de

[2] Fraunhofer AIS, Schloss Birlinghoven, D-53754 Sankt Augustin, angi.voss@fraunhofer.de

2.2 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.

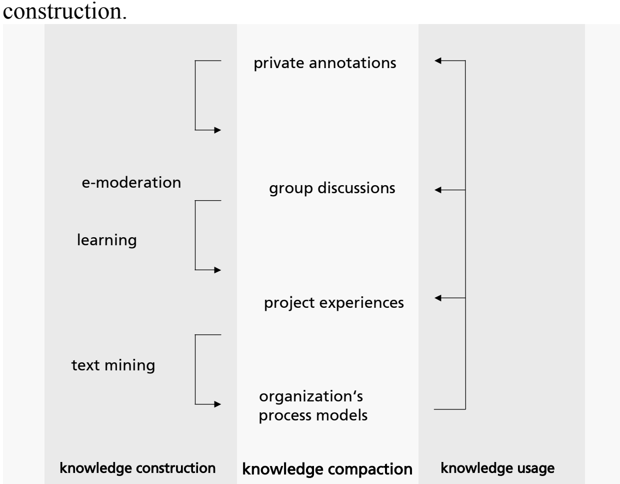


Figure 1. Layers of knowledge compaction, usage and creation for process-centered applications

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. Private 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 current 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.

On the one hand, indiGo is more comprehensive than approaches to experience management like (Althoff et al. 2001, Tautz 2000, Bergmann 2001, 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. On the other hand, indiGo is more focused than comprehensive approaches to organizational learning like ENRICH (Mulholland et al. 2000).

3 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 (Althoff et al. 1999).

Figure 2 shows the components of the indiGo platform as planned for the final version. This paper will focus on the version presented at CeBIT 2002, which comprises an integrator, CoIN-IQ, and Zeno.

---

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

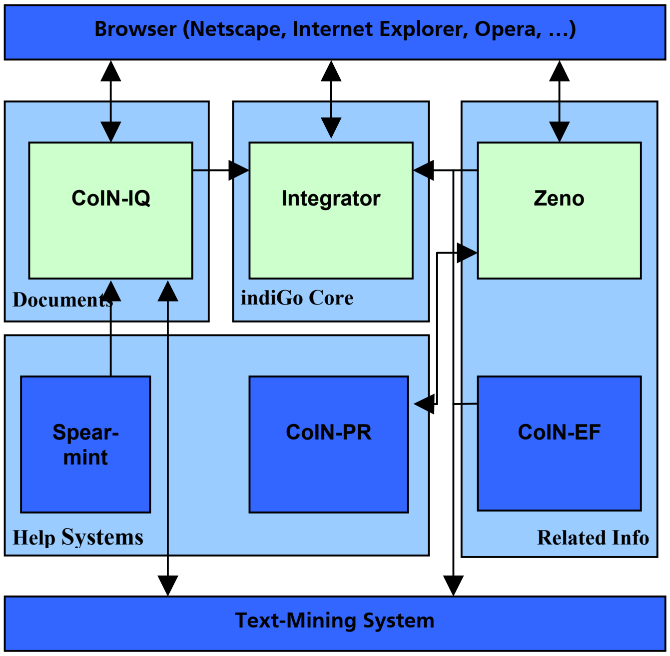


Figure 2. Information flow in the indiGo platform (upper level presented at CeBIT 2002)

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.

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.

3.1 The 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 information (see figure 3). Based on Perl the integrator is a CGI script that offers three fundamental functions that are called either by CoIN-IQ or Zeno:

- *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.

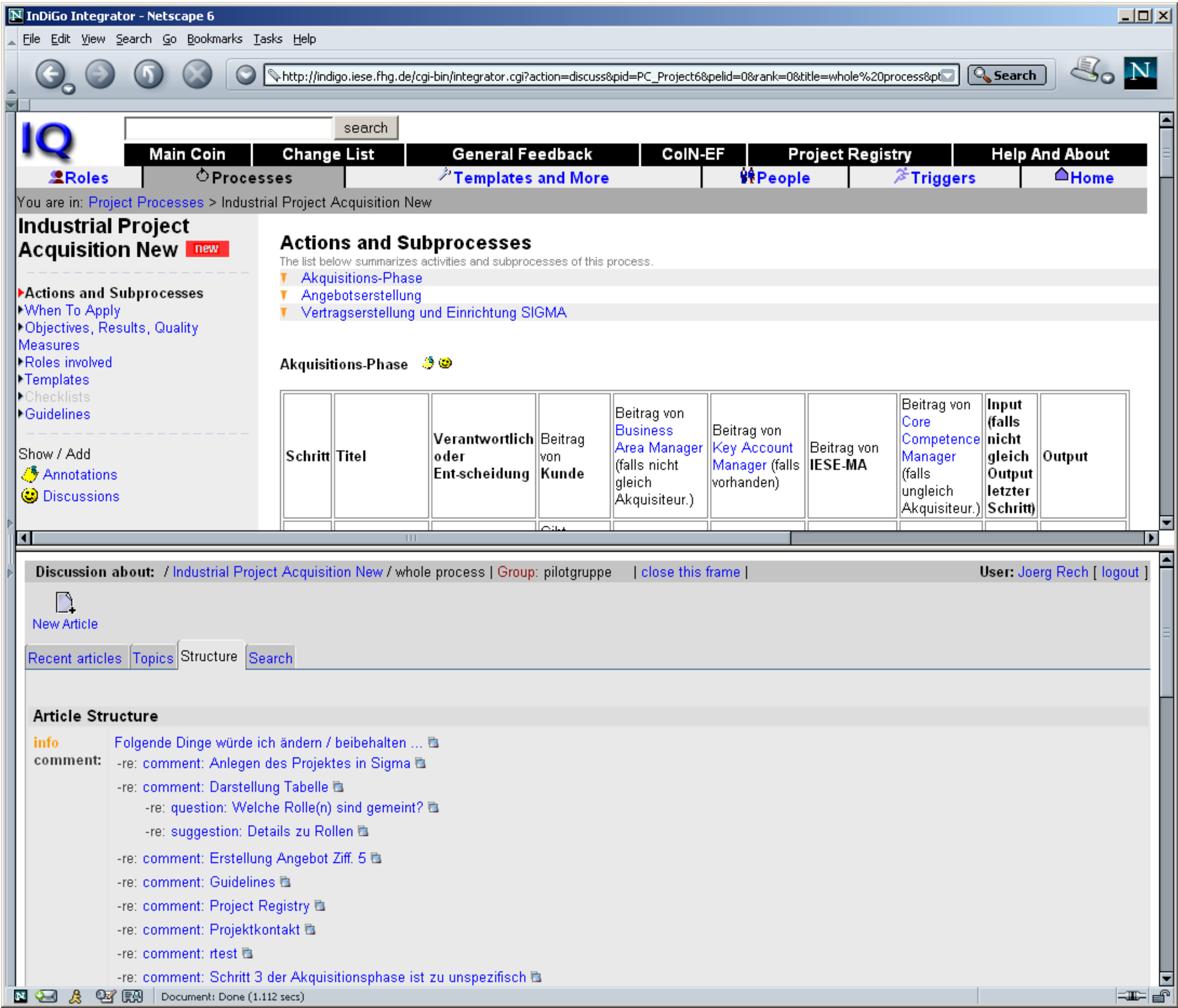


Figure 3. Split View with CoIN-IQ at the top and a related discussion in Zeno beneath

3.2 CoIN-IQ

CoIN-IQ is IESE's business process model repository (Decker and Jedlitschka, 2001). 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).

3.2.1 Baseline

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

For the general purpose of process models, Curtis, Kellner, and Over (1992) identify five different categories: Facilitate human understanding and communication, support process improvement, support process management, automate process guidance, and automate execution. According to this classification scheme, CoIN-IQ fits into the first category of facilitating human understanding and communication: The processes are executed by human agents (i.e., IESE members), based on the process description. Supporting and enforcing process execution beyond this human-based approach (e.g., by workflow modeling and enactment as in (Maurer and Holz 1999)) was regarded as non-suitable for the purposes of IESE due to the creative nature of its business processes. Furthermore, processes according to the process models are executed rather infrequently (< 10 times per month), therefore (a) automation of the processes was not supposed to leverage a high cost/benefit and (b) tracking of process status can be done by asking the responsible process executor. In addition, the experience made with the Electronic Process Guide (EPG) (Becker-Kornstaedt & al. 1999) showed that web-based process descriptions are a feasible way of distributing process knowledge within creative environments such as software business. In particular, changes to web-based process models can be communicated much quicker than paper-based process models, thus enabling quick integration of experience.

The origin of process models can be empirical (i.e., based on actual processes (Bandinelli, Fugetta et. al 1995)) and theoretical (i.e., reflecting a planned process execution). Process models in CoIN-IQ have both origins: Some of the process models reflect well-established processes (like, e.g., the administrative project set-up), others represent new procedures (e.g., the reflection of recent changes in the organizational structure of IESE).

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

The process modeling technique of CoIN-IQ is structured text, which is due to several reasons: Zero effort training, straightforward modeling, and perpetuation in industrial strength applications. Zero effort has to be spent on training, since any IESE member can read structured text without previous training. Furthermore, straightforward modeling means that any IESE members can model processes using structured text, if supported by guidelines and the CoIN team. This aspect is additionally fortified by the experience in scientific publishing of most IESE members.

3.2.2 Concepts

To achieve these objectives, the following information is captured within CoIN-IQ:

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

The discussions in indiGo are related to process descriptions, which consist of "Actions and Subprocesses", "When to apply?", "Objectives, Results, and Quality Measures", “Roles involved”, “Templates”, “Checklists”, and "Guidelines" (see Figure 4).

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

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

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

*“*Roles involved*”* provides an overview of the roles involved in the process and links the Role Descriptions. An experienced user can quickly find the Role Descriptions that are distributed within the “Actions and Subprocesses” and “Guidelines” Section.

*“*Templates*”* lists the products referenced by the process description. This overview is intended to support IESE members, who are accustomed to the process and just need quick access to artifacts.

*“*Checklists*”* is also intended for the experienced user. It summarizes important steps and results of the Process Description.

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

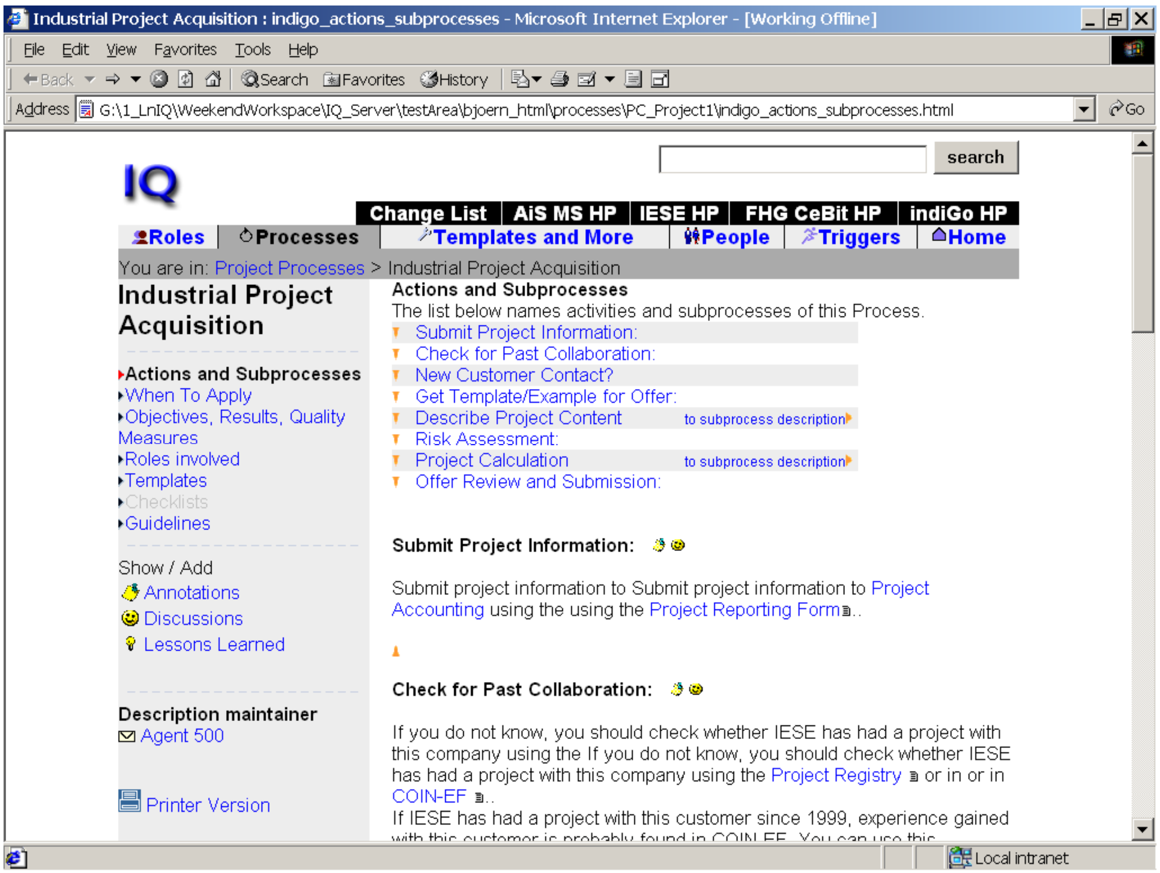


Figure 4. Screenshot of a process description. (Figure shows anonymized demonstrator)

3.2.3 Integration

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; this number will not change even if the process model is reorganized.

3.3 Zeno

Zeno is an e-participation platform (www.e-partizipation.org) (Voss 2002) with a spectrum of functions for moderated discourses on the web.

3.3.1 Baseline

Zeno comprises and extends (1) simple threaded discussions, (2) document-centered discourses, and (3) information structuring during group decision making.

Most electronic discussion forums, like the ones mentioned above but also newsgroups, support simple threaded discussions (1). Some tools, e.g. http://icommons.harvard.edu/, recognize URLs or even HTML tags in the contributions or allow to attach documents.

D^{3}E belongs to category (2). 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. D^{3}E is available as open source (http://d3e.sourceforge.net/) (Sumner & Buckingham Shum 1998). The e-learning platform 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 (3), especially for brainstorming (www.facilitate.com). Softbicycle's QuestMap (www.softbicycle.com) distinguishes questions, ideas, pros, cons, decisions, notes, and references, 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, that is, 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 that 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.

3.3.2 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 (see Figure 5).

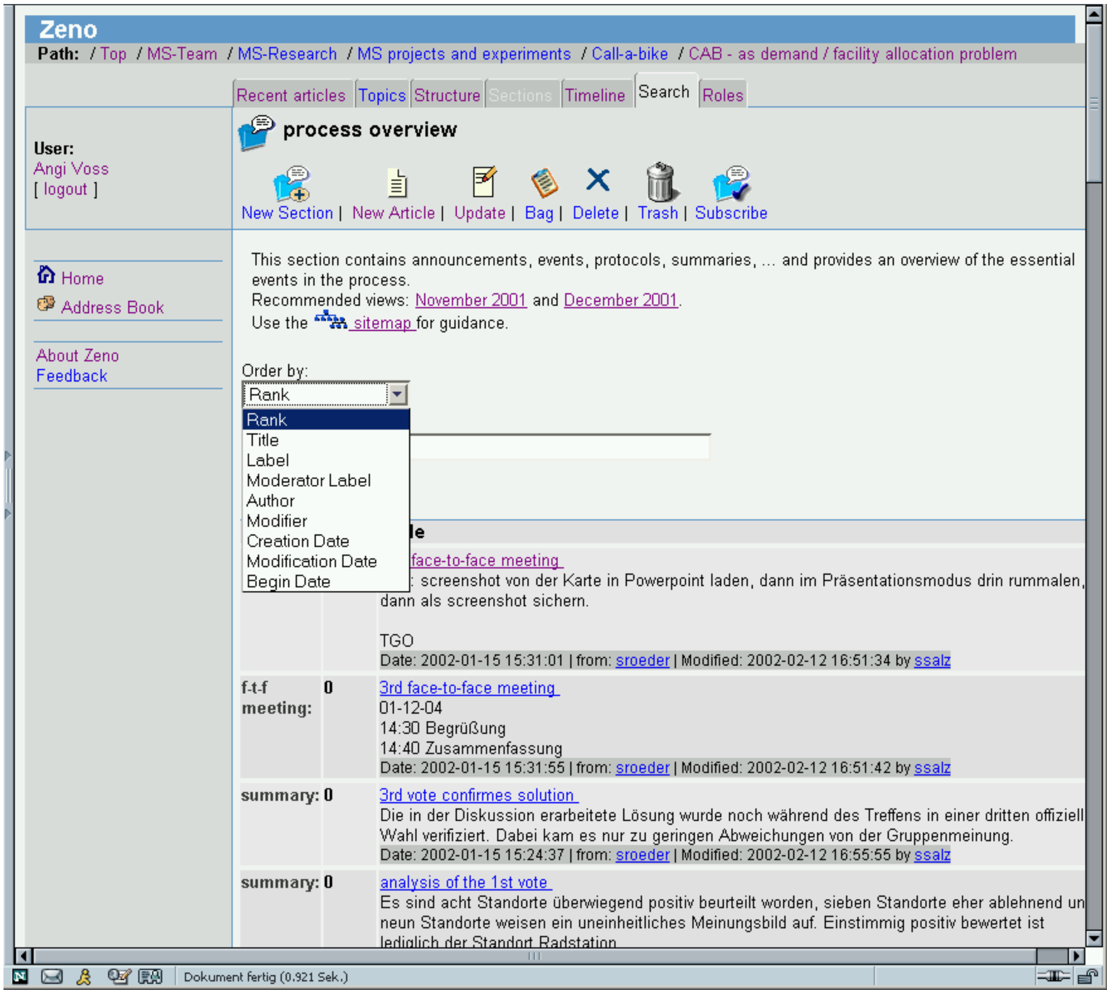


Figure 5. The search view in the overview section of a spatial decision making discourse in Zeno

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 accessed 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/).

3.3.3 Integration

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. 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.

5 CONCLUSIONS AND OUTLOOK

indiGo aims at supporting 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.

This paper focused on the technical infrastructure of indiGo, as presented at CeBIT 2002. It enables the organization of various process-related annotations and moderated discussion groups based on a customizable discourse grammar.

How an organization can accomplish process learning using the indiGo platform is the core of the indiGo methodology. In (Althoff et al. 2002) the methodology is described in more detail. It is itself phrased as a set of process models. The self-description of the indiGo methodology through indiGo process models offers the opportunity to ‘bootstrap’ indiGo, that is, to apply indiGo to itself. First, it allows having 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 perform and experience this approach, it will be their prime interest to resolve occurring difficulties. Therefore, the members of the organization can rely on a tested infrastructure and a consolidated team to support them in the roll-out phase.

In April 2002, the indiGo case study has been started, carried out at Fraunhofer IESE in Kaiserslautern, Germany. New project and research processes will be introduced for the whole institute. We expect very valuable feedback for all the described indiGo methods and technologies.

In parallel, work on the software platform is progressing with specified but not yet implemented features. 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. Next, the components indicated in Figure 2 will be integrated, starting with CoIN-EF.

indiGo’s e-moderation method guarantees that discussions are carried out 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.

As soon as discussions will become available from the case study, text mining experiments can begin (Kindermann et al. 2002, Leopold and Kindermann 2002). 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

Althoff, K.-D., Becker-Kornstaedt, U., Decker, B., Klotz, A., Leopold, E., Rech, J., Voss, A. (to appear 2002): "The indigo project: enhancement of experience management and process learning with moderated discourses". In Data mining in marketing and medicine (Ed. P. Perner), Springer Verlag, LNCS.

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.

Bandinelli, S., Fuggetta, A., Lavazza, L., Loi, M. & Picco G.P. (1995). Modeling and improving an industrial software process, Transactions on Software Engineering, 440-454.

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.

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

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

Curtis, B., Kellner, M. I., and Over, J. (1992). Process modeling. Communications of the ACM, 35(9):75-90, Sept. '92

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

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

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.*

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 & Diederich, Joachim & Leopold, Edda & Paaß, Gerhard (2002): *Identifying the Author of a Text with Support Vector Machines*; accepted at *Applied Intelligence*.

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.

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

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).

Mulholland, P., Domingue, J., Zdrahal, Z., and Hatala, M. (2000). Supporting organisational Learning: An Overview of the ENRICH Approach. *Journal of Information Services and Use*, 20 (1) 9-23.

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.

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.

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.

Inductive-Deductive Databases for Knowledge Management

Marcelo A. T. Aragão, Alvaro A. A. Fernandes
Department of Computer Science
University of Manchester
Oxford Road, Manchester M13 9PL, UK
{m.aragao|a.fernandes}@cs.man.ac.uk

Abstract. The growing awareness by organizations that knowledge is a key asset has led to an intense interest in tools that support knowledge management tasks. Ideally, such tools will support not only the discovery of knowledge by inductive exploration of data but also the fluent exploitation of that knowledge to produce actionable information. This paper expresses a position that a logic-based approach to the integration of knowledge discovery and deductive query answering offers significant advantages in terms of effectiveness and usability. Adopting a knowledge management perspective throughout, the paper describes an engine that exhibits integrated inductive and deductive inference capabilities and briefly considers the issues that arise in such an integration endeavour. A proof-of-concept implementation of the engine has been built and the paper uses it to suggest the potential benefits accruing from the position adopted. This is done by describing the deployment of the prototype in a classical knowledge management workflow. The paper aims to contribute an approach to logic-based knowledge management tools that have the potential for high levels of effectiveness and usability as a direct consequence of the uniformity in both the representations used and in the algorithmic treatment by means of which data, knowledge and information are made used of or derived from those representations. In this respect, the goal is to support in as fluent as possible a manner a more comprehensive set of knowledge management workflows than has hitherto been possible.

1 Introduction

The last few decades have delivered efficient, reliable and relatively inexpensive technologies for the management and exploitation of data stocks. This has helped consolidate the view of data stocks as primary assets of modern organizations. More recently, there has been a surge of interest in treating knowledge[1] too as a primary asset [6]. By definition, the main purpose of data and knowledge is to feed the processes comprising information production.

From this viewpoint, actionable information is the ultimate goal in the value-adding chain that has roots in data and knowledge stocks. Nonetheless, it is increasingly clear that, in most organizations, an ever larger proportion of the available data stocks lies largely unexploited. Correspondingly, most organizations now realize that the amount of knowledge they effectively exploit is far smaller than it needs to be if they are to remain competitive. Therefore, organizations are now more conscious than ever that valuable, actionable information might be waiting to be uncovered in the data stocks they already hold or can easily acquire.

Increases in knowledge stocks, on the other hand, are much harder to achieve, as technological support for that is not yet in place. This paper is motivated by the desire to contribute to the foundations of a class of knowledge management tools that are likely to prove particularly useful in this context.

Although database technology has delivered the means to manage and exploit data stocks and recent progress in knowledge discovery from databases justifies a certain degree of optimism, there is still no detailed proposal for an integrated platform for the combined management and exploitation of data and knowledge stocks. For instance, the link between knowledge discovery and knowledge management is very much under-explored [19]. This paper presents one such proposal, which we refer to as *inductive-deductive databases (IDDBs)*.

A proof-of-concept implementation of the algorithms and policies contributed by Section 4 has been carried out[2] and the motivating example discussed in Section 2 runs exactly as described. Note, however, that no claim is made that this prototype implementation is a contribution on its own. Rather, it has been built solely for the purpose of allowing the potential effectiveness and (in some respects) usability of IDDBs, as opposed to their potential efficiency, to be experimented with. This is precisely how the prototype is used in Section 2, viz., to provide more concrete motivation for the benefits that might accrue from the contributions of the paper.

The remainder of the paper is structured as follows. Section 2 uses a prototype implementation of the combined inference engine to present an extended example and show how the databases characterized in Section 4 constitute a step in the direction of providing suitable platforms for organizations that aim at faster rates of growth in their knowledge stocks.

[1] In this paper, by *knowledge* is meant *explicit knowledge*, i.e., knowledge that can be formally represented for use in information production processes.

[2] Available for download at `http://www.cs.man.ac.uk/~aragaom/iddb-1.tar.gz`

This section assumes the reader possesses basic knowledge of logic programming and deductive databases [3], including the syntactic conventions in those areas. Technical background is introduced in Section 3. Section 4 characterizes IDDBs by presenting the algorithms that model the operation of their combined inference engines. Section 5 discusses how the IDDBs defined in Section 4 can comprise more than a single inductive engine. Related work is discussed in Section 6. Finally, Section 7 points at both work that is underway and at future work that the contributions of the paper make possible while drawing a few conclusions stemming from the latter.

2 Deploying IDDBs in Knowledge Management

Data and knowledge stocks feed the processes of information production and must be refreshed, adjusted, adapted and, ultimately, increased if information stocks are to underpin future competitiveness in organizations. The simplified dataflow diagram in Figure 1 illustrates how diverse and interdependent data, knowledge and information management tasks are. One key challenge arising from the complexity illustrated in Figure 1 lies in devising platforms that can fluently support most of those tasks and foster purposeful growth and exploitation of all the stocks involved. This section exemplifies how IDDBs might be used in a range of tasks depicted in Figure 1 with levels of effectiveness and fluency that are a direct consequence of the uniformity in both the representations used and in the algorithmic treatment by means of which data, knowledge and information are made used of or derived from those representations.

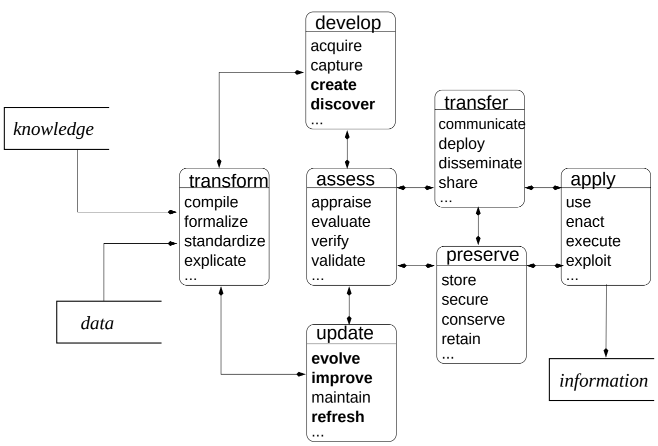


Figure 1. Information Management Activities (inspired by [11])

The authors have built a prototype IDDB to begin exploring the validity of this claim. The prototype implements the engine and policies formalized in Section 4. The remainder of this section consists of a summarized, step by step, description of how a workflow involving the exploitation and growth of data and knowledge stocks for the production of actionable information can be carried out using the prototype IDDB. Each step is annotated with reference to the tasks in Figure 1 that the step contributes to.

Step 1: [**preserve**] Suppose an organization records data about its employees (the convention is followed in the paper that metadata, e.g., schema expressions are signalled by an initial '`$` ').

```
$employee(<name>,<position>,<degree>,<graduate_from>).
$involves(<position>,<skill>).
 employee(mary, programmer, maths, oxford).
 ...
 involves(programmer, databases).
 ...
```

The fragment above shows that, for each employee, their name, position, degree, and the university from which they graduated is recorded. The skills (e.g., `databases`) that each position involves are also recorded.

Step 2: [**develop**] Assume now that the organization decides to implement a personnel policy of allocating employees to positions that minimize the need for training.

```
$suitable(<name>,<position>).
suitable(Employee,Position):-
  competent(Employee,Skill),
  involves(Position,Skill).
```

The policy can be roughly described by the rule above. It defines an employee as suitable for a position if s/he is competent in the skills involved. The rule is asserted in the intensional part of the database.

Step 3: [**develop, apply**] However, for this policy to be effective, one must be able to characterize the competences that employees possess. In particular, because the definition of how competences arise is taken to be complex and to suffer from volatility, the organization decides to characterize `competent/2` as an inducible concept (signalled by an initial '`&` '). There is a need, therefore, to induce a definition for it in terms of skills. To do so, examples are needed. So, a sample of employees is invited to reply to a survey about what skills they consider themselves to have. The results of the survey are used to augment with an additional attribute the data stocks about those employees that replied to the survey.

```
&competent(<name>,<skill>).
 competent(fiona, databases).
 ...
```

This can take the form of asserted positive examples of `competent/2` such as indicated above. Note that, if the induction algorithm cannot learn from positive data only (as is the case with mFOIL [7] used in the prototype IDDB built by the authors), a set of negatives examples can be methodically generated (as is the case in the prototype) by appealing to the closed-world assumption inherited by IDDBs from databases in general.

Step 4: [**develop, preserve, transfer**] Given the current background knowledge, the positive examples obtained from the survey and, possibly, negative examples valid under the closed-world assumption, the prototype learns the definition below for `competent/2` . The request for the definition to be learned is signalled by an initial '`?-` ' (as is a request for a query to be answered). The prototype responds with the definition of `competent/2` below.

```
?- competent(Employee, Skill).
```

```
competent(Employee,Skill):-
  employee(Employee,Position,_Degree,_Graduate),
  involves(Position,Skill).
```

Note that, of course, the prototype takes quite a simplistic view of how language bias is specified (as can be verified in Section 4). A more expressive implementation would allow more sophisticated mechanisms for specifying language biases, ideally with single-task granularity.

Note also that assuming, e.g., a policy for assimilation of inductive outcome that updates the knowledge stocks with the new definition and re-partitions appropriately the set of deducible and inducible predicates symbols, the stock of knowledge grows as a result of the system's having learned the definition above. Because of the uniform representation and algorithmic treatment, the new knowledge is fluently made available for information production.

Note, finally, that if, instead of submitting an explicit learning task as above, the user had requested the performance of the deductive task `?-suitable(Employee,Position)`, then, the definition above would be learnt on the fly, because the deductive evaluation of `suitable/2` depends on that of `competent/2`, so, if the latter lacked a definition one would be inductively derived there and then. In other words, the implemented prototype behaves lazily and suspends the deductive process to induce a definition for the required predicate and resumes the former once the latter has been obtained. In such cases, the user is warned that query answering required learning a new concept.

Step 5: [**develop**] Suppose now that a new-blood policy to hire recent graduates comes into force. The organization therefore announces several positions and the applications received are stored as indicated below.

```
$applicant(<name>,<degree>,<graduate_from>).
 applicant(zoe,cs,oxford).
 ...
```

Step 6: [**develop**] If the organization were to want to derive from the new data stocks information about the potential competences of the applicants so as to gauge to which positions they might be allocated, then the current definition of `competent/2` would not suffice. Since recent graduates are unlikely to have acquired the high-end skills that are targetted, changing the data capture is also not a solution. The organization therefore decides to refresh the current definition of `competent/2` and make it sensitive to the universities the applicants originate from.

The idea is to build upon traditional strengths of different institutions to infer that their graduates are likely to be competent in such areas as the institutions are strong in. For example, graduates in computer science from Stanford are likely to have high-end skills in database technology, and this data could be captured from a variety of sources (e.g., the Stanford CS web site) and stored as indicated below.

```
$strong_at(<graduate_from>,<degree>,<skill>).
 strong_at(stanford,cs,databases).
 ...
```

Step 7: [**update, develop, apply**] A new learning task can then be devised as follows. The example sets can be augmented by examples derived from the original applications of current employees. These employees and their respective skills are well-known in the organization, and an inductive task that takes those as positive instances will tend to reflect the organization's past decisions to hire.

The current definition for `competent/2` is dismissed by re-partitioning the set of inducible and deducible predicates so as to include `competent/2` in the former and `suitable`, `employee`, `involves`, `strong_at`, and `applicant` in the latter. Since the goal at this point in the workflow is to refresh the characterization of `competent/2` in the light of the enhanced background knowledge about the strengths of the applicant's academic roots, it is sensible to drop the previously learned definition. Then, submitting the inductive task again causes the knowledge about competences to be refreshed as per the new definition shown below.

```
?- competent(Employee, Skill).

 competent(Employee,Skill):-
   employee(Employee,Position,_Degree,_Graduate),
   involves(Position,Skill).
 competent(Applicant,Skill):-
   strong_at(Graduate,Degree,Skill),
   applicant(Applicant,Degree,Graduate).
```

Step 8: [**transfer**] Finally, assuming this definition to be retained, the answer to a query such as: `?-applicant(Applicant,_,_),suitable(Applicant, Position)` can be used as actionable information with which to allocate applicants to positions while respecting the policies in place. Therefore, in the example above, every time the allocation policy has to be applied, a deductive task can provide the necessary information. Correspondingly, every time the notion of competence, upon which an allocation policy is based, has to be developed, an inductive task can exploit the current stocks of data and knowledge available in order to create or update it.

Note that, data stocks have increased in Steps 3, 5 and 6. Knowledge stocks have increased in Step 2, by explicit assertion, and in Steps 4 and 7, by automated induction. Finally, information stocks have grown throughout, as both data and knowledge stocks are handled in an integrated manner. This extended example runs exactly as described in the proof-of-concept prototype that the authors have built and made available.

This section has exemplified the broad functionality of IDDBs with reference to Figure 1 in order to provide some evidence for the claim that IDDBs hold the promise of fluent and effective data, knowledge and information management. In the next two sections, first the background to IDDBs is presented and then IDDBs themselves are characterized in some detail.

3 Background

The representation language used in this paper is **Datalog**, described in detail in [3]. Recall that Datalog is a function-free Horn-clausal language, with well-understood model- and proof-theories. Datalog can be extended with negated atoms in the body (e.g., by stratification). This paper assumes the notion of a **deductive database** [3], defined to be a set S of

Datalog clauses partitioned into an intensional part (consisting entirely of safe rules – i.e., rules in which every variable occurring in the head occurs in at least one literal in the body), called the **intensional database** and denoted by IDB_S, and an extensional part (consisting entirely of ground facts) called the **extensional database** and denoted by EDB_S. For technical reasons, in a deductive database S, the predicate symbols occurring in the head of clauses in IDB_S are not allowed to occur in the head of clauses in EDB_S. Furthermore, queries can only involve predicate symbols in IDB_S. Therefore, querying EDB_S must be done indirectly. It is customary to assume that each predicate symbol in EDB_S has a corresponding implicitly-defined select-all view in IDB_S and not to distinguish two predicate symbols that are related in this way. Given a finite set S of Datalog clauses, recall that its **Herbrand universe** HU_S is the set of individual constants in S and its **Herbrand base** HB_S is the set of atoms formed from the predicate symbols in S and the terms in HU_S, and since S is finite, so are HU_S and HB_S. If S is a deductive database, then $EDB_S \subseteq HB_S$. A **query** Q_S over a deductive database S is a rule body formed with the predicate symbols in S. Classically, an **answer** A_{Q_S} to a query over a database is a set of substitutions of variables in Q_S with elements from HU_S so as to characterize atoms in the unique **least Herbrand model** LHM_S associated with S and Q_S. In this paper, the assumption is made that the subset of LHM_S corresponding to Q_S is returned rather than the substitutions that characterize them. Henceforth, subscripts are often omitted if the context is clear. The query evaluation process over deductive databases is tractable and has been extensively studied in both a bottom-up and a top-down direction [3]. In this paper, a query evaluation algorithm over deductive databases is assumed and referred to as deduce. Any one of many such algorithms described in the literature [3, 5, 13] may serve as a denotation for deduce. Thus, the remainder of the paper assumes to be well defined an expression such as $A := \mathsf{deduce}(Q, (IDB, EDB))$ that assigns to A the answer to a query Q over a deductive database $IDB \cup EDB$.

Inductive logic programming (ILP) [17] can be seen as an approach to the provision of inductive functionality that shares the underlying foundations of deductive databases. This paper confines itself to Datalog as the representation language and adopts throughout the example setting of ILP [17]. Thus, given a set B of clauses, taken to be **background knowledge**, and sets E^+ and E^- of **positive** and **negative examples**, respectively, such that some elements in E^+, and all elements in E^-, are false in LHM_B, the goal is to find a set H of clauses, referred to as the **hypothesis**, such that all elements in E^+ are true in $LHM_{B\cup H}$ and all elements in E^- are false in $LHM_{B\cup H}$. This concept learning process has been extensively studied in both a bottom-up and a top-down direction [17, 20]. In this paper, a concept learning algorithm over deductive databases is assumed and referred to as induce. Any one of many such algorithms described in the literature [10, 16] may serve as a denotation for induce. Thus, the remainder of the paper assumes to be well defined an expression such as $H := \mathsf{induce}(L, (B, E^+, E^-))$ that assigns to H a hypothesis conforming to the language bias L that, given background knowledge B, covers all the positive examples in E^+ and none of the negative ones in E^-.

4 Inductive-Deductive Databases

This section builds upon the wealth of research into deductive databases and into ILP to characterize a class of databases (represented as sets of Datalog clauses) that exhibits deductive and inductive inference capabilities.

Given deduce and induce functions as described in Section 3, the new functionalities provided by the class of databases contributed by the paper are embodied in an information-extracting algorithm that outputs data (represented as Datalog facts) by the application of deduce and knowledge (represented as Datalog rules) by the application of induce. The main challenge involved lies in devising and implementing a strategy to partition, in the dynamic way required, the data and knowledge stocks that are input to each inferential step. The main benefit derived is the integration of deductive and inductive inference into a continuum that underpins knowledge management workflows approximating the schema in Figure 1.

Let $\Pi = (\Delta, I)$ denote a partition of a set of predicate symbols (each with a fixed arity – denoted, for a predicate symbol p, by $\mathsf{arity\_of}(p)$) into two sets. The set Δ is the set of predicate symbols defined to be available for deduction as would be classically understood in a deductive database context. One can think of it as characterizing the relation names in a database schema, and hence, as the names of concepts in the belief set that underpins deductive processes. The set I is the set of predicate symbols targetted for induction as would be classically understood in language biases of ILP tasks. Let the set I of inducible predicate symbols in $\Pi = (\Delta, I)$ be denoted by $\mathsf{inducibles\_in}(\Pi)$. Finally, given a Datalog clause C of the form $A \leftarrow B_1, \ldots, B_n$, let $\mathsf{head\_of}(C)$ denote the atom A, and given an atom A, let $\mathsf{predicate\_of}(A)$ denote the predicate symbol of A.

An **inductive-deductive database** (IDDB) is a triple $S = (\Pi, K, D)$ where K denotes a stock of knowledge in the form of an intensional database, D denotes a stock of data in the form of an extensional database, and $\Pi = (\Delta, I)$ is a pair of predicate-symbol partitions as described above. Given a list $t_1, \ldots, t_n$ of atoms where each $t_i, 1 \leq i \leq n$, denotes either a deductive or an inductive task, an algorithm can be defined over an initial IDDB $S_0 = (\Pi_0, K_0, D_0)$ that, for each task in $t_1, \ldots, t_n$, effects the transition of S_{i-1} into a new IDDB $S_i = (\Pi_i, K_i, D_i)$, $1 \leq i \leq n$ and returns an outcome O, where O is the information derived from the task.

One such algorithm is given in Figure 2. It can perform a single task, as in lines 3 to 14, or it can perform task flows sequentially, as in lines 15 to 18. In this last case, the outcomes of each task are accumulated, as in line 18. The base case is trivial, as in lines 1 to 2.

The information derived by the algorithm is data (i.e., Datalog facts) if the task is deductive, and knowledge (i.e., Datalog rules) if the task is inductive. The decision as to whether it is a deductive task or an inductive one is made in line 4, according to the predicate partition in use. Hence, a deductive task is performed from lines 6 to 9, while an inductive task is performed from lines 11 to 14.

The algorithm in Figure 2 makes use of four auxiliary functions, which are better thought of as implementing policies regarding the user's view on the epistemological status of the set of concepts captured in Π. In particular, policies must

```
perform(Tasks,(Π,K,D)) ≡
    CASE (Tasks MATCHES [])
        THEN RETURN ((Π,K,D),∅)
    ELSE (Tasks MATCHES [Task])
        IF predicate_of(Task) ∉ inducibles_in(Π)
            THEN
                (IDB,EDB) := set_deductive_basis(Π,K,D)
                Outcome := deduce(Task,(IDB,EDB))
                (Π1,K1,D1) := assimilate_deductive_outcome(Task,(Π,K,D),Outcome)
                RETURN ((Π1,K1,D1),Outcome)
            ELSE /* predicate_of(Task) ∈ inducibles_in(Π) */
                (B,E+,E-) := set_inductive_basis(Task,(Π,K,D))
                Outcome := induce(Task,(B,E+,E-))
                (Π1,K1,D1) := assimilate_inductive_outcome(Task,(Π,K,D),Outcome)
                RETURN ((Π1,K1,D1),Outcome)
    ELSE (Tasks MATCHES [First | Rest])
        ((Π1,K1,D1),O1) := perform([First],(Π,K,D))
        ((Π2,K2,D2),O2) := perform(Rest,(Π1,K1,D1))
        RETURN ((Π2,K2,D2),O1 ∪ O2)
    ESAC
```

Figure 2. An Algorithm for Performing Tasks Over Inductive-Deductive Databases

be in place to determine at each inference step, what is to count as the basis for that inference step and what is to be done with its outcome. For example, a policy is needed to decide what, if any, induced knowledge can be used in subsequent deductive inferences. The spectrum varies from total permissiveness (i.e., all induced clauses are always unconditionally automatically assimilated into knowledge stocks) to total skepticism (i.e., no induced clause is ever automatically assimilated), with – as might be expected – intermediate, and more realistic and useful, points (e.g., automatically assimilate only those induced clauses that survive evaluation hurdles regarding one or more of accuracy, relevance, significance, justifiability, understandability, novelty, etc.).

The outcome of deductive inferences can also be the subject of a policy regarding assimilation. This might model materialization of queries, for example, although one would expect the default case to be no assimilation at all. More interestingly, a policy is needed to establish the basis for a deductive step. In particular, if clauses in the intensional database are allowed to refer in their bodies to predicates that need to be learned (e.g., for refreshment, assuming that the knowledge they capture is volatile), then one needs to consider inducing these before they can be used in answering a query. This can be done lazily, on an on-demand basis, regardless of whether the deductive engine proceeds top-down or bottom-up. Alternatively, one might want to cache (or even materialize) the entire (or selected portions of the) latent knowledge stock as lemmas.

For illustration purposes, Figure 3 presents example policies. With respect to what is to count as the basis for each deductive inference step, set_deductive_basis implements a policy of total inclusiveness and maximum recency, i.e., it always automatically exploits all the latest knowledge. Thus, a new definition for each predicate currently in inducibles_in(Π) is induced and unioned to the intensional database, as in lines 3 to 8, in order to avoid that a deductive task fails or blocks because one predicate lacks definition or has an outdated one. Although this policy may be onerous, it does allow for sequence, iteration, alternation and interleaving of deductive and inductive steps, thereby providing great flexibility to users. The pseudo-code in Figure 3 is at a level of abstraction higher than that in which lazy, on-demand induction of concepts might be specified, but the prototype implementation used in Section 2 deploys exactly such an implementation strategy. With respect to what is to be done with the outcome of a deductive inference step, assimilate_deductive_outcome implements a policy of zero assimilation, i.e., irrespective of the task and its outcome, the results of deductive inferences are never automatically assimilated, as in line 1 of Figure 3. With respect to what is to count as the basis for each inductive inference step, set_inductive_basis implements a policy suitable for concept description, i.e., taking as background the stock of current data and knowledge, modulo what is targetted for induction, in line 1, as positive examples any facts that are known to be instances of the target concept, in line 2, and as negative examples, appealing to the closed-world assumption [3] that deductive databases abide by, (possibly a subset of) the facts in the Herbrand base of the concept that are not known to be instances of the concept, in line 3. With respect to what is to be done with the outcome of an inductive inference step, assimilate_inductive_outcome implements a policy of total permissiveness, i.e., irrespective of the task and its outcome, the results of inductive inferences are always automatically unconditionally assimilated, i.e., the target concept is removed if it exists, in lines 1 and 2, and the new definition is inserted, in lines 3 and 4. Note that no claim is made that any of these policies are in any way guaranteed to have desirable consequences only. Clearly, many issues are raised that demand further investigation. It is important to stress that policies are, by definition, user-specific and the granularity with which they are put in place needs to be carefully thought through in each case. Options vary from very fine (e.g., a task may be accompanied by a policy that is set or chosen to hold with scope bound to that task alone) to very coarse (e.g., policies are hardwired into the implementations). It may also be the case that policies are devised by, and put in place for, groups of, rather than individual, users. This flexibility can be explored, e.g., in incremental data mining and in deploying sophisticated knowledge assimilation techniques.

set_deductive_basis(Π, K, D) $\equiv$

  EDB := $\{ x \in D \mid$ predicate_of(head_of(x)) $\notin$ inducibles_in(Π) $\}$
  IDB := $\{ x \in K \mid$ predicate_of(head_of(x)) $\notin$ inducibles_in(Π) $\}$
  FOR EACH Π IN inducibles_in(Π) /* assuming arity_of(Π) $= n$ */
    BEGIN
      B := $IDB \cup EDB$
      E^+ := $\{x \mid x \in$ deduce($\leftarrow \pi(V_1, \ldots, V_n)$, ($K$, D))$\}$
      E^- := $\{x \in HB(K \cup D) \mid x \in HB(\{\pi(V_1, \ldots, V_n) \leftarrow\}) \wedge x \notin E^+\}$
      IDB := $IDB \cup$ induce($\pi(V_1, \ldots, V_n)$, (B, E^+, E^-))
    END
  RETURN (IDB, EDB)

assimilate_deductive_outcome($Task$, (Π, K, D), $Outcome$) $\equiv$

  RETURN (Π, K, D) /* e.g., independently of $Task$ and $Outcome$ */

set_inductive_basis($\pi(V_1, \ldots, V_n)$, (Π, K, D)) $\equiv$

  B := $\{ x \in K \cup D \mid$ predicate_of(head_of(x)) $\notin$ inducibles_in(Π) $\}$
  E^+ := $\{x \mid x \in$ deduce($\leftarrow \pi(V_1, \ldots, V_n)$, ($K$, D))$\}$
  E^- := $\{x \in HB(K \cup D) \mid x \in HB(\{\pi(V_1, \ldots, V_n) \leftarrow\}) \wedge x \notin E^+\}$
  RETURN (B, E^+, E^-)

assimilate_inductive_outcome($\pi(V_1, \ldots, V_n)$, (Π, K, D), $Outcome$) $\equiv$

  K_1 := $K \cup \{ x \in Outcome \mid$ predicate_of(head_of(x)) = Π $\}$
  D_1 := $D \setminus \{ x \in Outcome \mid$ predicate_of(head_of(x)) = Π $\}$
  I_1 := inducibles_in(Π) $\setminus \{\pi\}$
  Δ_1 := ($\Pi \setminus$ inducibles_in(Π)) $\cup \{\pi\}$
  RETURN ((Δ_1, I_1), K_1, D_1) /* e.g., new knowledge persists unconditionally */

Figure 3. Example Policies for Exploiting and Assimilating Data and Knowledge

4.1 The IDDB Prototype

A prototype of an inductive and deductive database was implemented in Prolog. The prototype implements the algorithm in Figure 2 and, with additional performance improvements, the policies in Figure 3 on top of a service layer that enables storage and retrieval of Datalog clauses. For the deductive function, it uses the deductive database described in [5], while, for the inductive function, it uses the implementation of mFOIL described in [10].

The architecture of the prototype separates, via well-defined call interfaces, inference functions and policies, so that their replacement can be carried out non-disruptively and all other parts of the algorithms remain unchanged. Section 5 shows how this can be done.

For instance, different inductive algorithms may be used interchangeably provided that they adhere to the interface defined by the function induce.

The prototype was tested with the example in Section 2, and shown to be more effective than the separate application of SLD-resolution and mFOIL, because it enables an increase in data and knowledge stocks which are not comparably achievable otherwise. For example, a deductive task may derive answers that are only obtainable if some concept is induced on-the-fly (in which case the epistemological status of any output is clearly signalled).

The prototype qualifies as a proof-of-concept that combining deductive and inductive inferences is effective. Moreover, its development demonstrates that combining inference modes need not be more complex than the inference functions and policies already are. This is the central insight borne out by the implementation of the IDDB prototype described in this paper.

Although the primary goal is not efficiency and scalability, the prototype brought to light interesting issues. Firstly, combining deductive and inductive inference does not necessarily increase the overheads, in terms of time complexity, inherent to the inference functions. Secondly, this combination would not scale up if inputs to and outputs from the inference functions were passed by copy, thus requiring that larger amounts of data (or knowledge) be kept and copied in memory. The algorithm in Figure 2 is defined in this way, but only for clarity of exposition. The prototype addresses scalability issues more efficiently. The inference functions were wrapped in such way that its parameters and temporary results can be accessed on demand, so as to make the prototype as scalable as the inference functions it instantiates. Finally, improvements in the inference functions and policies are likely to propagate to the combined inference database engine, as illustrated above by the use of lazy evaluation to implement a sophisticated preparation policy for deduction. Based on this, it is possible to claim that the efficiency and scalability obtained for the combination of inference modes is not necessarily worse than for the implementations of the inference functions individually. Therefore, the choice of good implementations for the inference functions and policies is likely to pay off in terms of efficiency and scalability.

Another feature of the prototype is that data and knowledge are assigned an explicit epistemological status so that its use during inference is kept coherent. The prototype maintains a partition of the set of predicate symbols to control this, and it is this partition that is allowed to vary from task to task.

Further work is needed, and indeed is planned, to explore in detail both the abstract and the concrete issues arising from the ideas presented in this paper.

5 Extending IDDB Engines

An extended version of the IDDB prototype was implemented which, in addition to concept learning (based on mFOIL), also supports conceptual clustering. Support for clustering was included without affecting the existing functionalities, due to the modular architecture of the prototype.

This was done by assuming the expression $T :=$ cluster$(F, (B, C))$ to be well defined which, given background knowledge B, conforming to the features defined in F and the set of instances in C, assigns to T the conceptual clusters derived with respect to the similarity function provided by the algorithm that instantiates it. Then, the algorithm in Figure 2 was extended to output conceptual clusters (represented as Datalog rules) by applying the inference function denoted by cluster, as follows:

1. another partition is introduced where elements are the predicate symbols targetted for clustering, i.e., containing the names of relations whose instances may be clustered;
2. preparation and assimilation policies are introduced as the functions set_clustering_basis and assimilate_clustering_outcome, respectively. For example, set_clustering_basis may generate an axiomatization for the relations is_a and a_kind_of, that can be used as background knowledge, while assimilate_clustering_outcome may transform and assimilate the clustering information as a set of Datalog clauses; and
3. a clustering algorithm is selected that implements the function cluster. The implementation of COBWEB [9] used is by Joerg-Uwe Kietz, available publically in [14].

This extension represents an improvement in terms of usability for the prototype, in the sense that it supports knowledge discovery via supervised or via unsupervised learning. Thus, a broader class of knowledge management applications can be supported.

In order to evaluate the extended prototype, it was applied to one of such application, i.e., the construction and use of taxonomies. A Lotus Discovery Server [4] tutorial also uses this application to illustrate its potential as a knowledge management platform.

A taxonomy can provide organizations with a common business language and can serve as a navigational aid to finding business information. This allows its workers to drill down through abstraction levels until they find a class that describes information they need. In this example, key words and expressions (e.g., obtained from internal documents, white papers and web pages via information retrieval techniques) are represented extensionally as terms in Datalog facts that stand for the source documents. The preparation policy selects this set of Datalog facts and generates an axiomatization for the relations is_a and a_kind_of. Both are passed as inputs to the conceptual clustering algorithm (as, respectively, the set of instances and the background knowledge), while the clustering features are passed as arguments at task submission time. The assimilation policy converts the output of the clustering algorithm into Datalog rules that represent, declaratively, the structure and behaviour of the induced taxonomy. Then, the extended prototype can support, via deductive inference, browsing taxonomy hierarchies and classifying new instances accordingly.

In this context, the extended prototype was able to emulate the Lotus Discovery Server in building a working taxonomy from key expressions extracted from documents. No claim regarding accuracy or quality of the taxonomy derived is made here, because the extended prototype is based on a simpler algorithm for clustering, viz., COBWEB. Nevertheless, the extended prototype seems to provide tangible benefits in terms of fluency and effectiveness that would otherwise be unlikely to accrue from non-logical approaches.

Benefits regarding fluency are noticeable when specifying both taxonomy construction and exploitation as inferential tasks, hence, allowing them to be submitted in flexible and varied ways, as required. The prototype can support these tasks seamlessly. This may be useful for building taxonomies on demand in specific circumstances, either incrementally, so as to reduce engineering effort, or periodically, for refreshing taxonomic knowledge.

Benefits regarding effectiveness are noticeable when combining taxonomic knowledge with existing knowledge stocks because both are uniformly represented as Datalog clauses. There are also benefits in exploring taxonomic knowledge through deductive inference, since recursive, ad-hoc queries can be used to relate concepts that are not explicitly asserted in the taxonomy. These features would be hard to support with less expressive mechanisms without significant additional programming effort.

6 Related Work

The contributions of this paper are motivated by ideas stemming from the intersection of concerns such as integrating more tightly databases and data mining, providing a scalable platform for knowledge discovery in the large, extending databases with the ability to perform inductive inferences, and others. Common to these research areas is the need to integrate the querying of models of data and the induction of such models from data. If one views querying as deduction, the way is open for a logic-based approach to integrating acquisition and exploitation. This paper follows more closely the application-driven view taken in [8]. In fact, the contributions of this paper constitute a detailed and concrete, albeit preliminary, exploration of the issues raised in [8].

Note that, despite the natural interest in doing so, for reasons of space, this section is silent about initiatives in which tools are brought together but not integrated at the representation level nor at that of the core engine (e.g., a data mining tool, such as Darwin, and a database management system, such as Oracle).

A few other proposals to extend database technology with inductive capabilities have shaped, or are related to, the contributions of this paper. The first such proposal introduced the idea of *inductive database-relations*. In [1], some of the relations over a deductive database may be left undefined and uninstantiated. This is similar to the partitioning in this paper of the predicate symbols into a set of inducible and a set of deducible ones. Little consideration is given in [1] to certain issues that this paper addresses in some detail, e.g., how to adjust data and knowledge stocks in the wake of, possibly interleaved and implicit, deductive and inductive steps. For this reason, while [1] has been inspirational, the similarities of that work with the one reported here do not run deep.

Another system that bears some resemblance to the work described here is Mobal [15]. Mobal can be seen as a knowledge acquisition environment that brings together several inductive logic programming schemes into an integrated whole and provides sophisticated services, such as theory restructuring, that the engine described in this paper is silent on.. While the overall functionality delivered by Mobal is impressive, it is also fixed and closed insofar as it is hard-coded behind a user interface. Mobal is not an instance of a database system (e.g., it lacks bottom-up query evaluation and integrity constraint mechanisms) and is not flexible enough that it can be adapted to be deployed as one. In contrast, IDDBs are database systems. They are also so designed as to be easily and cleanly extended. Thus, endowing IDDBs with more than just a single induction scheme is straightforward as discussed in Section 5. It is difficult to judge the degree of effort required to extend Mobal with the database features that it currently lacks. A final point of contrast is that while this research is based on Datalog, the logic underlying MOBAL goes beyond the tractability boundaries that Datalog was carefully designed not to cross. The practical implications of this fact could be significant, insofar as, while there is no meta-logical impediment for the database engines envisaged in this report to be efficiently implemented, the opposite can be said of systems such as MOBAL that are based on more expressive logics.

Recon is a data mining system described in [18]. The architecture of Recon includes a deductive database, a rule induction component (which outputs deductive database clauses) and a visualization component. Data is stored in relational databases and SQL is used to retrieve the data used as the inductive basis for the rule induction component. The rules generated in this way can then be tested against the source database and against the knowledge stored in the deductive database. This means that rules can be refined before being assimilated into the deductive database for further use. Visualization plays an important role in refining discovered knowledge. Query results can also be materialized temporarily for improved performance. Recon's main contribution is an environment that allows an interactive discovery and refinement of knowledge before assimilation, if needed. Recon, like MOBAL, offers a fixed and closed set of functionalities that are hard coded behind a user interface, hence, Recon exhibits many of MOBAL's shortcomings. The user must intervene explicitly to move permanent data to temporary stores before knowledge discovery can be performed. Discovered knowledge is also placed in temporary stores before it can be assimilated. Therefore, it is up to users to manage the flow of data and knowledge between physical stores, and this can compromise the fluency with which they can perform complex task flows.

Seminal ideas on *inductive databases* were proposed in [2, 12]. The contributions of this paper differ from those in that they are formulated from a logical perspective in which databases are seen as sets of logical clauses, whereas the inductive databases of [2, 12] conceive of knowledge stocks as patterns, more generally, and not logical clauses, more specifically, as this paper does. Also, [2, 12] make explicit an evaluation function on acquired knowledge and endow the former with queryable status. Such an evaluation function could be easily incorporated into the engine described here and making it queryable would be a means to provide evidence upon which assimilation polices might be configured. Another crucial distinction is that while this paper uses Datalog as its representation language, [2, 12] leave the latter unspecified, with most examples taking the form of propositional association rules. While the approach described here stands upon a well-defined and well-behaved logical framework, it is not clear at this stage what foundations the *inductive databases* proposed in [2, 12] stand upon.

7 Future Work and Conclusions

Work is already underway to extend the contributions described here in two main directions. On the other hand, the authors are also finalizing a generalized framework for specifying different policies for assimilation and exploitation of inductive and deductive outcomes. This will allow IDDBs to be seen as a class of systems whose instances can be determined by particular choices of inferential capabilities on the one hand, and assimilation and exploitation policies on the other.

Future issues that will be explored include studying, the formal, as well as empirical, motivations for different assimilation and exploitation policies; exploring the benefits of a tightly-coupled approach in which the two inference engines are subsumed by a unified one; and extending the range of knowledge discovery tasks beyond the modelling stage (to include, e.g., data preparation and model evaluation).

Preliminary though it is, this paper's characterization of a database platform integrating query answering and induction is nevertheless more detailed and concrete than any other past attempt with similar aims and scope. In particular, issues regarding the dynamic epistemological status of subsets of clauses are clearly highlighted here for the first time. The paper makes it clear that studying and developing policies for the assimilation and exploitation of the outcome of both deductive and inductive tasks is likely to be a major issue, albeit one that no previous related work has frontally addressed. For example, depending on the expressiveness of the logical language used, intermediate learning steps may introduce inconsistencies that are avoided in the case of IDDBs by sticking to (potentially stratified) Datalog.

To conclude, the contributions of this paper can be summarized as follows:

1. a formal characterization of IDDBs as a class of Datalog-based logical structures that subsume deductive databases;
2. an algorithm for performing tasks over IDDBs that fluently integrates deductive and inductive inference capabilities based on logic programming;
3. a characterization of the issues arising in the context of attempts to exploit and assimilate induced knowledge in knowledge management workflows;
4. an algorithm that embodies simple policies for exploiting and assimilating data and knowledge as an example of how the broader issues raised in this respect might be tackled;
5. an extended example of how IDDBs could deliver effectiveness and usability in practical knowledge management situations.
6. a description of how IDDBs can comprise more than one inductive framework.

Acknowledgement Marcelo A. T. Aragão is on leave from the Central Bank of Brazil and gratefully acknowledges the

scholarship awarded to him by the Department of Computer Science of the University of Manchester.

References

[1] Francesco Bergadano, 'Inductive Database Relations', *IEEE TKDE*, **5**(6), 969–971, (1993).

[2] Jean-François Boulicaut, Mika Klemettinen, and Heikki Mannila, 'Querying Inductive Databases: A Case Study on the MINE RULE Operator', in *Proc. 2nd PKDD*, eds., J. Zytkow and M. Quafalou, volume 1510 of *LNCS*, pp. 194–202. Springer, (1998).

[3] Stefano Ceri, Georg Gottlob, and Letizia Tanca, *Logic Programming & Databases*, Springer, 1990.

[4] Lotus Corporation. Lotus discovery server. more information available at `http://www.lotus.com/products/discserver.nsf`.

[5] Subrata Kumar Das, *Deductive Databases and Logic Programming*, Addison Wesley, 1992.

[6] Thomas H. Davenport and Laurence Prusak, *Working Knowledge*, HBS Press, 2000.

[7] Sašo Džeroski and Ivan Bratko, 'Handling noise in inductive logic programming', in *Proceeding of the 2nd International Workshop on Inductive Logic Programming*, ed., Stephen Muggleton, Tokyo, Japan, (June 1992). ICOT. TM-1182.

[8] Alvaro A. A. Fernandes, 'Combining Inductive and Deductive Inference in Knowledge Management Tasks', in *Proc. 1st Workshop on the Theory and Practice of Knowledge Management*, pp. 1109–1114. IEEE Press, (2000). In Proc. 11th Intl. Workshop DEXA.

[9] D. H. Fisher, 'Conceptual Clustering, Learning from Examples, and Inference', in *Proceedings of the 4th International Workshop on Machine Learning*, (1987).

[10] Nada Lavrač and Sašo Džeroski, *Inductive Logic Programming: Techniques and Applications*, Ellis Horwood, 1994.

[11] Ann Macintosh. Adaptive workflow to support knowledge intensive tasks. `http://www.aiai.ed.ac.uk/~alm/KAMSLIDES/index.htm`, September 1998.

[12] Heikki Mannila, 'Inductive Databases and Condensed Representations for Data Mining', in *Proc. 1997 ILPS*, ed., J. Maluszynski, volume 13, pp. 21–30. MIT Press, (1997).

[13] Jack Minker, 'Logic and databases: A 20 year retrospective', in *Proceedings of the International Workshop on Logic in Databases (LID'96)*, volume 1154 of *Lecture Notes in Computer Science*, pp. 3–57, San Miniato, Italy, (July 1996). Springer-Verlag.

[14] MLnet. Resource: Software. available at `http://www.mlnet.org/`.

[15] K. Morik, S. Wrobel, J.-U. Kietz, and W. Emde, *Knowledge Acquisition and Machine Learning - Theory, Methods, and Applications*, Academic Press, 1993.

[16] Stephen Muggleton, 'Inverse Entailment and Progol', *New Generation Computing*, **13**, 245–286, (1995).

[17] Stephen Muggleton and Luc de Raedt, 'Inductive Logic Programming: Theory and Methods', *JLP*, **19**(20), 629–679, (1994).

[18] Evangelos Simoudis, Brian Livezey, and Randy Kerber, 'Integrating inductive and deductive reasoning for data mining', in *Advances in Knowledge Discovery and Data Mining*, eds., Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, 353–373, MIT Press, Cambridge, (March 1996).

[19] Eric Tsui, Brian J. Garner, and Steffen Staab, 'The role of artificial intelligence in knowledge management', *Knowledge Based Systems*, **13**(5), 235–239, (May 2000).

[20] Stefan Wrobel, 'Inductive Logic Programming', in *Principles of Knowledge Representation*, ed., G. Brewka, 151–189, CSLI Pub., (1996).

Traceability and knowledge modelling

Smain Bekhti, Nada Matta

Tech-CICO, Université de Technologie de Troyes
12 rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France
e-mail: {smain.bekhti, nada.matta}@utt.fr

Abstract. A project memory is a representation of the experience acquired during projects realization. It can be gotten through a continuous capitalization of the enterprise activity, notably its design rationale. Most of capitalization methods don't allow a design rationale structuring in real time. We propose in this paper, a dynamic process of knowledge modelling, offering a way to keep track of Knowledge in two stages: direct transcription and structuring.

Keywords. Knowledge modelling, project memory, knowledge management, knowledge representation, design rationale.

1 INTRODUCTION

Knowledge management is a process of explicitation, modelling, sharing and appropriation of knowledge [1]. The majority of knowledge management methods aim at defining a corporate memory considered as a strategic asset of the organization. We can classify these methods in two main categories: knowledge capitalization methods and direct extraction methods (Figure 1).

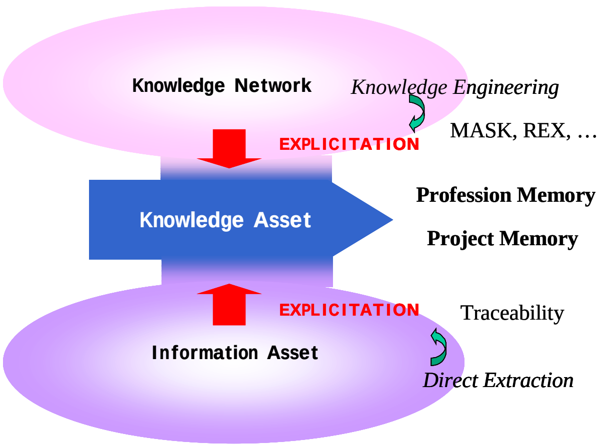


Figure 1 . Two techniques of explicitation of knowledge: capitalization and direct extraction

- The methods of knowledge capitalization use primarily techniques of knowledge engineering. These techniques consist mainly of knowledge extraction (experts interviews or collection from documents) and modelling. We can note for instance methods MASK, REX, etc.
- The direct extraction aims at extracting knowledge directly from the activity of the organization. We can distinguish several techniques as data mining (extracting knowledge using statistical analysis), text mining (extraction of knowledge based on linguistic analysis of texts [2]), techniques of traceability (e-mail, forum of discussion, etc) and design rationale.

We study in this paper, the traceability of the design rationale that aims at defining a project memory [3]. The principal problem in this traceability is the dynamic modelling, in other terms, how to formalize the data and information extracted in real time from the activity.

Dynamic modelling must also be realized in parallel with the organization activity. Therefore, this modelling should be integrated in this activity. In other terms, direct extraction and dynamic knowledge modelling introduce changes in the organization and the realization of a project.

Several methods of design rationale were defined. These methods allow keeping track of collective problem solving, especially those extracted in meetings of decision-making. The techniques recommended in these methods induce a consequent work. So they are less and less used in the organizations. The objective of our work is to define a method of dynamic modelling easy to apply, therefore a method easily integrated in the activity of realization of project. Our hypothesis is a decomposition of modelling in several stages, slightly transforming the activity of making notes and their organization. The method (Cf. 3) we defined is built by analysing an experiment of traceability of a project of definition of professional risks evaluation (in collaboration with National Institute of Research and Security «INRS» [4]) while being based on a study of the literature of the design rationale(Cf. 2).

2 TRACEABILITY OF THE DESIGN RATIONALE

Several methods were defined to represent the design rationale in a project. These methods can be classified in two principal categories: decision-making driven representation and problem solving dynamics representation.

2.1 The decision-making driven representation

In this type of approach, the design rationale, also named the analysis of the Space of design [5] is represented through the elements that influenced a decision-making. We can distinguish primarily the methods IBIS [6], DRAMA [7] and QOC [8] (the reader can refer to [3] to have more details about these methods).

The space of design is generally represented in these methods by design choices. These choices are structured like answers to the questions evoked by the design's problem. Arguments can justify the choices of an option according to a given criteria. The options generate other questions to which the designers answer by options.

2.2 Representation of the dynamics of problems solving

Some approaches offer a more global representation of the design rationale. Indeed, some elements of the context like the activity of the organization, the role of the actors and the artefact are represented. We can distinguish in particular the DRCS system [9]. It offers several views on a project: modules of the artefact, association of the tasks, evaluation of the specifications, decision-making, alternatives of design and argumentation.

Another approach consists of representing the design rationale based on cognitive analysis of a problem solving. We distinguish in particular DIPA formalism [10]. This formalism (Data, interpretations, proposals, agreement) use problems solving modelling defined in knowledge engineering to structure a decision-making. In DIPA, the model decision-making is represented in three major stages:

1. A first phase of description of the problem which allows collecting data, considered as symptoms in analyse situations or as needs in synthesis situations;

2. A second phase of abstraction which starts from data problems in order to find to them an interpretation corresponding to a possible cause in the analysis situations or with a functionality of solution in the synthesis situation;

3. A third phase of implementation which starts from the interpretation (cause or functionality) and which allows to elaborate a proposition which will take the form of a repair removing the cause of the symptom (analysis) or a means responding to the expressed functionality (synthesis).

2.3 Discussion

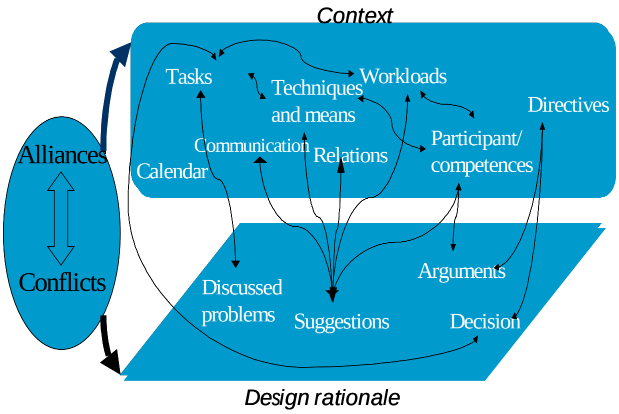


Figure 2 . Mutual influences between elements of the project

A project memory must contain elements of the experience Coming as well as from the context and from the problem solving. These elements have a strong mutual influence so that if the context is omitted, the restitution problems solving is insufficient.

We often observe this type of phenomena in the results obtained with the approaches quoted above. Except the system DRCS, some approaches defines techniques to represent this influence between the context and problems solving in a project. Even DRCS system can only allow representing a part of this context (the tasks organization and the projection of the decisions on the artefact). In the same way, we can observe some efforts in DIPA formalism to represent the organization of work in a workflow (task/role). However, also other elements have to be identified like constraints, directives, resources and competences, modes of communication, etc. We consider in our approach representing a complete vision of the project context by emphasizing its influence on the problems solving.

In other way, the representation of the problems solving as it is suggested by the approaches noted above, remains incomplete as a representation of the space of negotiation between the project actors. Indeed, the first type of approaches rather allows a representation driven by the decision in order to show only the elements that influenced a decision. In the second approaches type, an effort is made to represent the dynamics of the decision-making. However, a negotiation is a space of discussion between several actors where various objectives are confronted, alliances and conflicts are constituted. In the same way, a negotiation has a history and is influenced by the alliances and the decisions made during the last negotiations. Our approach permits to keep in memory this dynamics of negotiation so that its restitution is easy to show the various elements included in a resolution of problem.

Finally, the application of the design rationale methods proved their difficulties in real time. In fact, it is no evident to note all the enunciations and to analyse and structure discussion directly during the meeting. Modelling a-posteriori presents a significant risk of missing arguments and elements that influenced the decision-making. We propose in following, an approach proceeding by progressive stages for a direct traceability and a modelling of the negotiation.

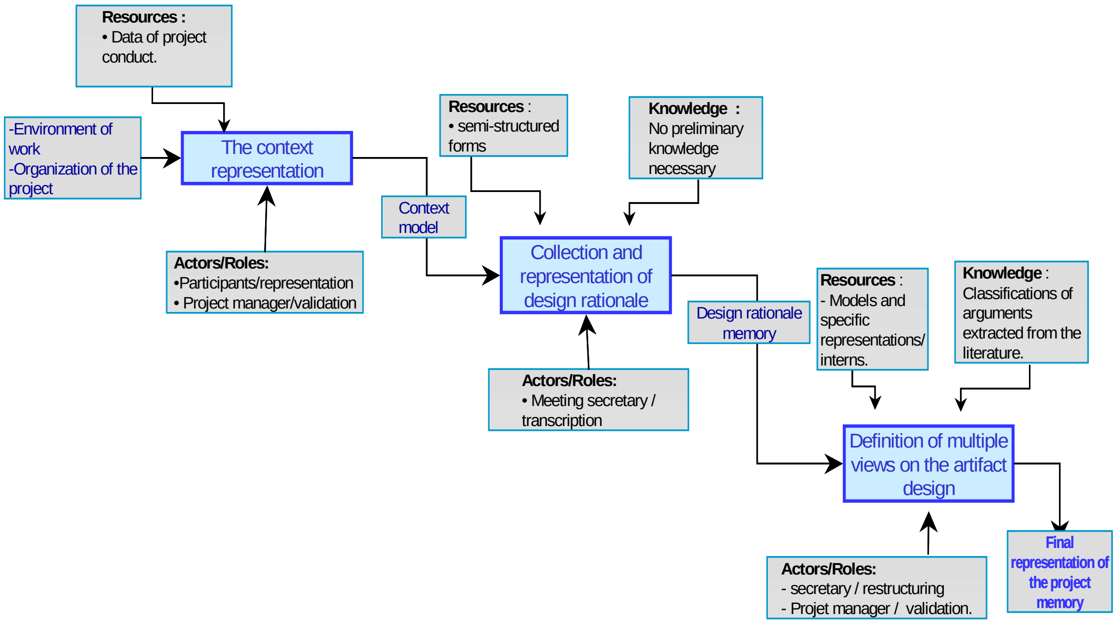


Figure 3 . Model of traceability process

3 DYNAMIC PROCESS OF KNOWLEDGE MODELLING

The dynamic process of knowledge modelling we defined is based on a method permits to obtain a structured track of a project memory as well as the context and decision-making. The principal objectives of the method are on the one hand, to make possible its application in real time and keep track of meeting and on the other hand to structure knowledge extracted so that it can be easily reusable. We thus defined three principal stages in this step: context representation, transcription of the design rationale, restructuring and multiple views definition (figure3).

3.1 Context representing

We represent the context of a project (Figure 4) as a description of the work environment (means and techniques, referential, instructions and constraints of the project) and the project organization (participants, their roles and tasks organisation).

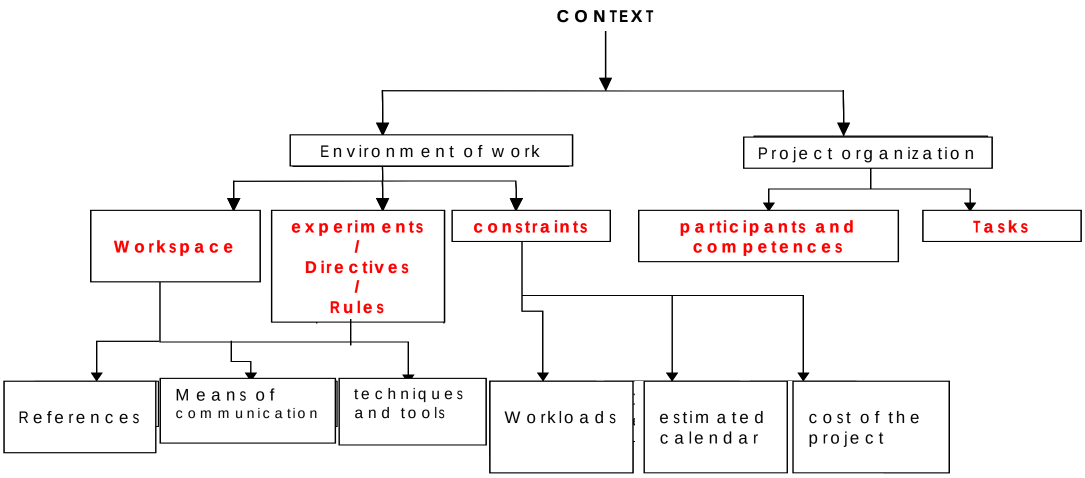


Figure 4 . Representation of the context

We present in figure4 some elements of the context that can be represented in several levels of structuring, to show different aspects of influence between its elements and the design rationale.

3.2 Extraction and representation of the design rationale

3.2.1 Direct transcription

The approaches of design rationale generally require a deep analysis to model decision-making. So they are not easily applicable in real time. The first stage of our approach consists of a transcription guided by a form where the basic elements as problems, argumentation and decision can be classified. These forms can be used to note in a structured and rapid way all the data elements that can be collected during a negotiation (Figure 5). The objective is to prepare a structured transcription of the negotiation during meetings and in real time. The structure of these forms permits to distinguish the elements of the discussed problem, to highlight the arguments of the participants to the meeting and their possible suggestions.

Notes are structured initially by participants who, during the meeting, are recognized either by their names or by their visual aspects. In fact, the direct transcription that we propose, follows on the one hand, the traditional methods of notes taking in meetings and on the other hand prepares the structuring of knowledge.

This transcription can be easily realised by a meetings secretary. No deep analysis is required in this type of transcription. Note also that a chronological recording of the negotiation is backed up in this type of transcription.

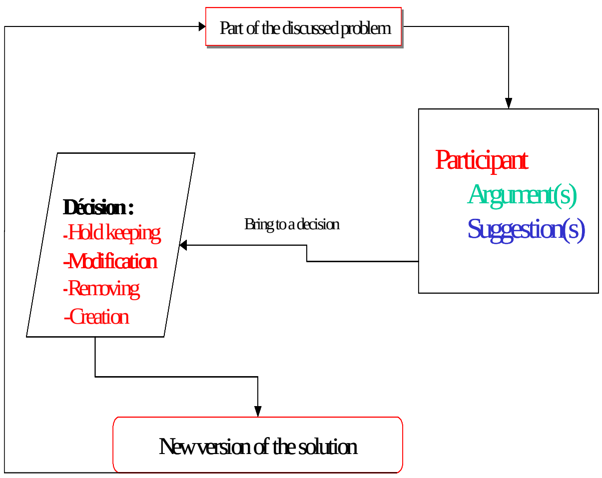


Figure 5 . Form used for the direct transcription of a negotiation

3.2.2 Content structuring

The principal objective of a structuring is to allow an intelligent access to the knowledge of the memory. We propose to provide several accesses to the memory according to various prospects that we define later on. The second stage of our approach consists of a structuring based on a cognitive analysis of the forms filled out during the direct transcription. We were inspired by the approaches of design rationale to define a structure of representation (Figure 6) putting ahead the influence elements of a negotiation, such as argument, criteria of justification and suggestion. The identification of the criteria is guided by a classification of the argument types. The method that we propose can be compared with meetings reporting where the direct transcription is similar to the notes making and the structuring to the summary report. However in our case, the notes taking is guided and the result is richer and reflects a more complete memory of the negotiation and the decision-making.

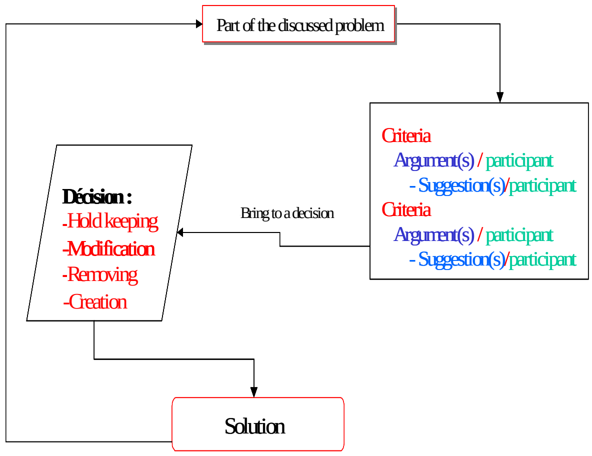


Figure 6 . Form used for the negotiation structuring

Some criteria, definite during this structuring, can be regarded as simple to identify and could be used to enrich the structure of the direct transcription (used in future meetings) and to facilitate the structuring. It's in this sense that we consider our method like dynamic process acting at the same time on the method and the structure.

Our main objective is to integrate the traceability of decisions in the process of realization of projects. The approach that we propose introduces a slight change into the organization of a project in order to make this traceability possible.

In order to guarantee a representation of the deep knowledge which have influence the design rationale, the validation meeting after some project phases and at the end, especially with some participants who have got a global vision of the project (for example, the project manager), must be hold. These permit to reformulate the arguments, the suggestions and the criteria and to re-examine their classification. The structure of the memory encourages the participants to clarify their knowledge, enriching by that the contents of the memory.

3.2.3 Logic of the structuring form

The structure represents the logic of discussion. Participants discuss each part of the problem by giving their opinions supported by several types. The participants can also give suggestions concerning the part of the problem. The whole arguments and suggestions allow the group to make a decision concerning this part of the problem. The part of the problem is thus solved, otherwise it will be discussed again in the same manner and it will pass by the same cycle. So we will be able to see the evolution of this element during the discussion until its final version.

In the structure, the arguments are classified according to their type or their nature. Each argument or suggestion is related to the participant who emitted it. Knowing that for each participant his competence and his role are described, that permits to see the relation that can exist between the contributions (arguments, suggestions) of the participants and their competence.

Elements of the structure

Problem objects: The global problem discussed during the meetings is composed of sub-problems or elements of problem. The idea is to break up the whole discussion into basic elements. The structure thus permits to represent these elements of discussion with their contents, to bind between them and to represent the evolution of each of them during the negotiations.

Arguments: One of the most significant elements of any negotiation is the argumentation. In our approach the argumentation is an essential element of the representative structure because it is the origin and the cause of the evolution of the discussion of the problem and consequently of the decision-making.

Suggestions: The arguments advanced by the speakers during meetings often lead them to make their own suggestions concerning such or such part of the discussed problem, we envisaged in the model a space for the suggestions of the participants. The suggestions are related to the arguments and the participants who proposed them.

Participants: The representation of the participants in the structure is important, it permits to bind the arguments and suggestions to their transmitters. Each participant is characterized, primarily, by his competences and his role in the project (see context). It permits to really understand the logic and the reasoning of the participants and the motives of their interventions.

3.3 Definition of multiple views

The design rationale as it is generally defined, represents the space of decision in a project. We propose to describe this space in various points of view while focusing on the negotiation that takes a central place in the design rationale. The majority of these points of view can be generated automatically from structuring forms. We identified four points of view: Point of view of problem solving, Point of view of argumentation criteria, Point view evolution of the problem solving and chronological point of view. We study other points of view that permit to shows the links between the participants and the problem solving [11].

3.3.1 Point of view of problem solving

This point of view is based primarily on the structured forms corresponding to the elements of the problems treated.

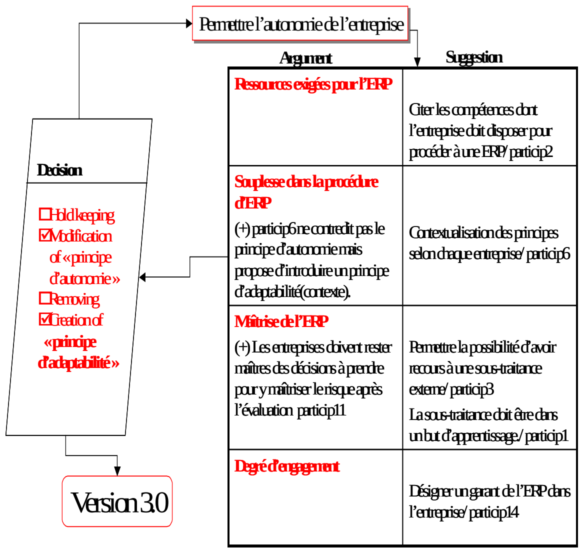


Figure 7 . Example of a point of a view on the problem solving [4].

3.3.2 Point of view of argumentation criteria

A view extracted from the criteria of argumentation shows a synthesis of the key elements that influenced the problem solving and from through that the decision-making. This view presents the relations between the criteria, the advanced arguments and the arising problems.

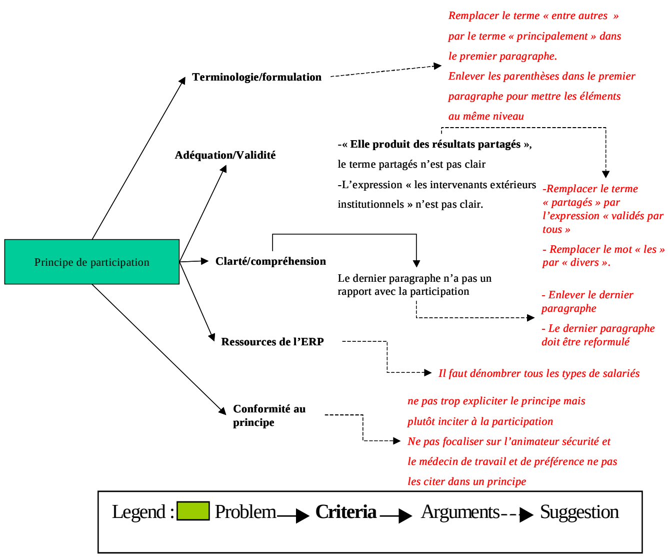


Figure 8 . Example of a view by criteria

3.3.3 Point of view evolution of the problem solving

The evolution of the decisions is an important element to memorize in the design rationale. We put the evolution of the problems forward while joining the problem to its solution that can also generate other problems.

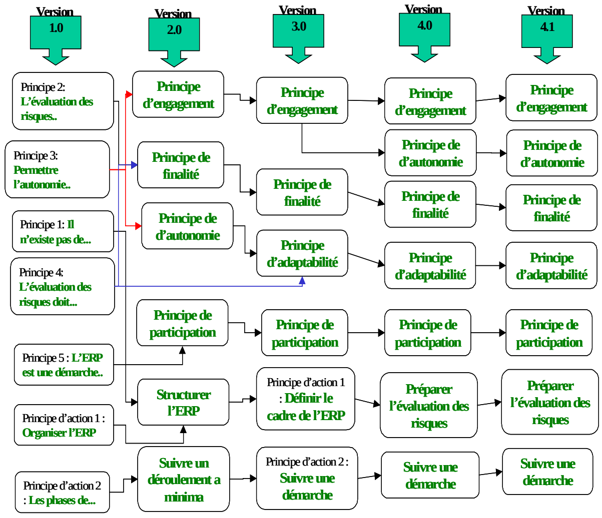


Figure 9 . Example of a view on the evolution of the artefact (principles of assessment of the professional risks).

3.3.4 *Chronological point of view*

The transcription forms can offer a chronological view on the progress of the negotiation. Indeed, from this chronological representation, we can reach at any phase of the evolution of the problem solving. The representation of the task process in the context as well as the link between these tasks and the forms provide a global view on the progress of the project.

4 CONCLUSION

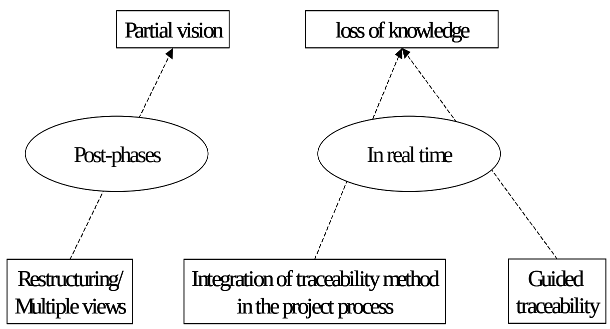


Figure 10 . The solutions brought by our approach to the limits of traceability methods

A project memory reflects an acquired experience, it must represent all elements of information related to the project, as well as the context and the design rationale. We describe in this paper an approach that permits a global representation of these elements. It puts forward the elements and the mutual relations that influence the problem solving in a project and that through views representing the different faces of the project progress.

The approaches of traceability of the design rationale present some limits in the modelling during the activity. These limits are linked essentially to the difficulty in identification and classification in real time questions, suggestions, types of arguments, etc. during meeting. We proposed a dynamic process of modelling based on several phases starting from a semi-structured note taking toward a more advanced structuring. The structure of representation evolves the problems evolution.

Our approach is based on a representation similar to the approaches of design rationale. Indeed, the decision-making is described with key words as: problem, arguments, suggestions, etc. As we showed it in this paper, it integrates easily in the project process without requiring specific expertises. It is based on as well as knowledge traceability in real time and a-posteriori analysis that permits to get a deep representation of knowledge. Thus, allows having a global vision of the project (figure10). Let's note that the process of modelling is based on an abstraction guided by classifications and structures.

We defined this approach while being based on a real experience (the project of definition of the principles of assessment of the professional risks) and we plan to validate it on other fields of application.

The representation of the context in our approach is not developed enough, we examine other studies of the context especially mathematical and sociological representation. The pragma-linguistic works can enrich the representation of the communication in a memory of project, in the same way, the socio-organizational studies are very important to identify the interpersonal relation and their role in the decision-making.

We develop a tool to support our approach offering, on the one hand, a flexible structure of representation and on the other hand an adaptive user interface.

5 RÉFÉRENCES

[1] Dieng R., Corby O., Giboin A. et Ribière M. – Methods and Tools for Corporate Knowledge Management, in Proc. of KAW'98, Banff, Canada. 1998

[2] Bourigault D. et Lépine P. – Utilisation d'un logiciel d'extraction de terminologie (LEXTER) en acquisition des connaissances, Acquistion et Ingénièrie des Connaissances, tendances actuelles, Editions Cépaduès, 1996

[3] Matta, N., Ribière, M., Corby, O., Lewkowicz, M., et Zacklad, M. Project Memory in Design, Industrial Knowledge Management - A Micro Level Approach. SPRINGER-VERLAG : RAJKUMAR ROY, 2000

[4] Bekhti S., Matta N., Andéol B. et Aubertin G. – Mémoire de projet : Processus dynamique de modélisation des connaissance , CITE'2001, p. 329-345. Troyes, 29-30 Novembre 2001

[5] Buckingham Shum S. – Representing Hard-to-Formalise, Contextualised, Multidisciplinary, Organisational Knowledge. Proceedings of AAI Spring Symposium on Artificial Intelligence in Knowledge Management, P.9-16, 1997. http://ksi.cpsc.ucalgary.ca/AIKM97/AIKM97Proc.html

[6] Conklin J.E. et Begeman M.L. – gIBIS: A Hypertext Tool for exploratory Policy Discussion, ACM Transactions on Office Informations Systems, 6,303-331, 1998.

[7] Brice A. – Design Rationale Management (DRAMA), http://www.quantisci.co.uk/drama

[8] MacLean A., Young R.M., Bellotti V.M.E., Moran T.P., – Questions, Options, and Criteria: Elements of Design Space Analysis, Human-Computer Interaction, Vol.6, 1991.

[9] Klein M. – Capturing Design Rationale in Concurrent Engineering Teams, IEEE, Computer Support for Concurrent Engineering, January 1993.

[10] Lewkowicz M., Zacklad M., MEMO-net, un collecticiel utilisant la méthode de résolution de problème DIPA pour la capitalisation et la gestion des connaissances dans les projets de conception, IC'99, Palaiseau, p.119-128. 14-16 juin 1999,

[11] Brown David C., Berker I., – Modeling Conflicts Between Agents in a Design Context, Computational conflicts, Conflicts Modeling for Distributed Intelligent System, 144-164, Springer 2000.

Knowledge Management Performance Index Considering Knowledge Cycle Process

Kun Chang Lee[1], SoonJae Kwon[2], Namho Chung[2], Soochoul Joung[2], Byung-uk Kang[2]

Abstract. This paper is aimed at proposing a new metric named KMPI (Knowledge Management Performance Index) to evaluate the performance of knowledge management (KM) at a point in time. Firms are assumed to have always been oriented toward accumulating and applying knowledge to create economic value and competitive advantage. On the basis of this assumption, we suggest KMPI, a new metric having a logistic function with five components of knowledge circulation process (KCP)- knowledge creation, knowledge accumulation, knowledge sharing, knowledge utilization, and knowledge internalization. If KCP efficiency increases, then KMPI will become greater, which means that firms are now becoming knowledge-intensive businesses. To prove the contribution of KMPI more formally, questionnaire survey was conducted extensively among 101 firms listed in KOSDAQ market in Korea, and we associated KMPI with three financial measures such as stock price, PER, and R&D expenditure. Statistical results show that the proposed KMPI can represent the KCP efficiency, and go along with the three financial performance measures.

Keywords: *Knowledge management performance; Knowledge circulation process; Logistic function; KMPI; Factor analysis*

1. INTRODUCTION

A knowledge-based view of the firm, which has emerged as one of important strategic management topics, provides theoretical basis about why knowledge-based resources are playing an important role in increasing sustainable competitiveness of the firm (Cole, 1998; Spender, 1996ab; Nonaka and Takeguchi, 1995). The resource-based view of the firm suggested by Penrose (1959), Barney (1991), Teece (1998), and Wernerfelt (1984) promotes the knowledge-based perspective of the firm, which postulates that competitive advantage builds upon those privately developed resources, tacit and explicit, developed inside the firm. Likewise, the knowledge-based view of the firm posits that the knowledge assets existing at any given time per se, one of those idiosyncratic resources proprietarily created and accumulated in the firm for years, produce sustainable competitive advantage. In this new era of highly competent IT, this knowledge-based view of the firm can explain convincingly why certain firms show more competitiveness under the same market situation. The knowledge assets are dependent upon the quality of organizational knowledge and intangible assets in general (Grant, 1996ab). Even though we adopt the knowledge-based view of the firm, there exists an important research question- why do most firms that initiated KM still struggle with the development of appropriate metrics to assess the effectiveness of their initiatives. In other words, they need some metrics to justify their KM initiatives financially. They don't want to make their KM look like pure research activity that may some day lead to remarkable increase in management productivity and performance. In any case, linking KM initiatives to important financial measures may help to justify KM investments to senior management and more importantly improve the firm's ability to manage knowledge assets effectively. Given that several KM benefits are intangible, one measurement method that is growing in popularity is the balanced scorecard (Kaplan and Norton, 1992). Alongside financial measures, the balanced scorecard includes other perspectives, i.e., customers, internal business processes, as well as innovation and learning. However, linking KM initiatives to performance measures both tangible and intangible is not enough. We need a more rigorous measurement metric to assess the KM performance with an ability to explain it and suggest future strategic movement the firms should take to improve their KM performance. To address this research question and need, our research objective is to propose a new measurement metric, named KMPI (Knowledge Management Performance Index), to evaluate KM performance. Basic assumption underlying KMPI is that knowledge may be viewed from an unified perspective – it has multi-faceted characteristics like a state of mind (Schubert et al., 1998), an object (Carlsson et al, 1996; McQueen, 1998), a process (Zack, 1998), a condition of having access to information (McQueen, 1998), a capability with the potential for influencing future action (Carlsson et al, 1996; Watson, 1999). Alavi and Leidner (2001) summarized well the distinction between each perspective about knowledge. Table 1 is an excerpt from p.121 in Alavi and Leidner (2001).

Table 1. Diverse perspectives of knowledge and their implications for KM (Excerpt from Alavi and Leidner (2001), p.111)

Perspectives		Implications for KM
State of mind	Knowledge is the state of knowing and understanding	KM involves enhancing individual's learning and understanding through provision of information
Object	Knowledge is an object to be stored and manipulated	Key KM issue is building and managing knowledge stocks
Process	Knowledge is a process of applying expertise	KM focus is on knowledge flows and the process of creation, sharing, and distributing knowledge
Access to information	Knowledge is a condition of access to information	KM focus is organized access to and retrieval of content
Capability	Knowledge is the potential to influence action	KM is about building core competencies and under standing strategic knowhow

[1] Professor of MIS, School of Business Administration, Sung Kyun Kwan University, Seoul 110-745, Korea, leekc@skku.ac.kr
[2] Graduate School of Business Administration, Sung Kyun Kwan University, Seoul 110-745, Korea

Table 2. Five categories of KM studies

Category	Implications	Sub-categories	Researches
General	Several managerial and social issues pertaining to KM are dealt with.	KM strategy and organizational culture	Pentland (1995), Alavi and Leidner (1999), Zack (1999)
		Specific processes and activities within KM	Petrash (1996), Szulanski (1996), Alavi (1997), Elliott (1997), Van der Spek and Spijkervet (1997), Choo (1998), Holsapple and Joshi (1999)
		Review and research agenda	Davenport and Glover (2001), Gold et al. (2001), Alavi and Leidner (2001)
Learning organization	Firms maintain organizational knowledge to obtain a sustainable competitive advantage.	Organizational knowledge	Stata (1989), Senge (1990), El Sawy et al. (1986), Schatz (1991-1992), Stein and Zwass (1995), Walsh and Ungson (1991), Tuomi (2000), Markus (2001)
		Learning capability and design of leaning organization	Purser et al., (1992), Roth and Senge (1996), Van de Ven and Pooley (1992), Shaw and Perkins (1992), McGill and Slocum (1994), Leonard-Barton (1995)
Knowledge assets evaluation	Valuing and measuring intangible assets promotes organizational learning and generates organizational capabilities.	Intellectual capital	Brooking (1996), Edvinsson (1997), Sveiby (1998)
		Balanced Score Card	Kaplan and Norton (1992)
		Strategic organizational learning and organizational capabilities	Massey et al. (2001), Roos and Roos (1998), Sakaiya (1991), Stewart (1997), Teece (1998, 2000), Leonard and Sensiper (1998)
Role of IT	KM should be supported by IT and/or KMS so that KM can contribute to increasing management performance.	Knowledge Management System (KMS)	Alavi (1997), Alavi and Leidner (1999, 2001), Baird et al. (1997), Bartlett (1996), Davenport et al. (1996), Gray (2000), Henderson and Sussman (1997), Rouse et al. (1998), Sensiper (1997), Watts et al. (1997)
		Role of IT in KM in general	Alavi and Leidner (1999), Newell and Scarbrough (1999), Pérez-Bustamante (1999)
		Role of IT for specific KM activities	Liou and Nunamaker (1993), Khalifa (1998), Fischer et al. (1999), Petraglia and Glass (1999), Squires (1999), Suthers (1999)
		Knowledge mining and DSS for KM	Rouse et al. (1998), Holsapple and Joshi (2001)
		Strategic use of the Internet	Dieng (2000), Martin and Eklund (2000), Dominique and Motta (2000), Schwartz and Te'eni (2000), Rabarijaona et al. (2000), Szykman et al. (2000), Caldwell et al. (2000)
Success and failure factors	Success factors for KM should be given sufficient consideration before launching KM strategy.		Davenport et al. (1996), Ruggles (1998), Glasser (1999), Krogh (1998)

Based on the unified perspective of knowledge, this paper posits the following four assumptions without loss of practicality and theoretical generality: (1) KM activities may be reduced into performing knowledge circulation process (KCP) in which there exist five components making KM operational in the firm- knowledge creation, knowledge accumulation, knowledge sharing, knowledge utilization, and knowledge internalization.
(2) KM is defined here tactically by all kinds of management activities promoting KCP in a firm.
(3) A firm can increase its flexibility and adaptability to rapidly changing business environment by focusing on the efficiency of KM activities.
(4) Within the firms adopting KM, KMPI will gradually increase with time.
The layout of the paper is as follows. In the following section, we describe the previous studies to justify our research objective. Section 3 explains fundamentals and theoretical backgrounds of the proposed KMPI. Research hypotheses are suggested. Then the empirical study is suggested with promising results, in Section 4. In conclusion, this paper is ended with discussing contributions of this research and future research directions.

2. PREVIOUS STUDIES

Previous studies on KM builds on multiple disciplines, e.g., management, computer science, and information systems. To maintain consistency in our literature survey on KM and to justify our research objective, i.e. to suggest a new measurement metric named KMPI for assessing the KM performance, we classify previous KM literature into five categories- (1) general, (2) learning organization, (3) knowledge assets evaluation, (4) role of IT, (5) success and failure factors, all of which are summarized in Table 2.

First category deals with managerial and social issues related to KM. Some studies (e.g., Pentland, 1995, Alavi and Leidner, 1999, Zack, 1999) stressed the importance of the strategy driving KM and the organizational culture within which KM takes place. Other studies focused directly on specific processes and activities within KM, e.g., knowledge acquisition, generation, storage, distribution, application and measurement (Petrash, 1996; Szulanski, 1996; Alavi, 1997; Elliott, 1997; Van der Spek and Spijkervet, 1997; Choo, 1998; Holsapple and Joshi, 1999). Also, research agenda and general perspective of KM based on extensive literature review are presented (Gold et al.,2001; Davenport and Grover, 2001; Alavi and Leidner, 2001).
Second category takes management perspective that deals with questions about what learning organization means for obtaining a sustainable competitive advantage. Especially, according to Stata (1989) and Senge (1990), learning is the only sustainable competitive advantage for organization, and learning ends up with leaving organizational knowledge (or memory) (El Sawy et al., 1986; Schatz, 1991-1992; Stein and Zwass, 1995; Walsh and Ungson, 1991). Markus (2001) shows a theory of organizational knowledge reuse. A sustainable competitive advantage may be obtained through organizational knowledge which is a corporate memory having an effect on present decisions and playing as an important factor in the success of an organization's operations and responsiveness to the changes and challenges of environment (Stein and Zwass, 1995; Walsh and Ungson, 1991). There have been increasing attempts to help organizations improve their learning capability and to design themselves as learning systems (Purser et al., 1992; Roth and Senge, 1995). In addressing how organizations can improve their learning capability, researchers have identified a number of problems that organizations face when trying to learn (Van de Ven and Pooley, 1992; Shaw and Perkins, 1992; McGill and Slocum, 1994; Leonard-Barton, 1995).
Third category is concerned with evaluation of knowledge assets. Practicing KM for years can produce various forms of intangible assets or intellectual capital within firms. Such intangible assets are intellectual capital. Edvinsson (1997) shows, based on case study of Skandia, that the intellectual capital of a firm can be measured, documented, and monitored. Brooking (1996) analyzes the multiple components of intellectual capital and provides lists of high-level questions useful for auditing an organization's intellectual capital. In addition, Sveiby (1998) details how to effectively use and measure intangible assets and how to monitor them for financial success. Kaplan and Norton (1992) develop Balanced Score Card (BSC) using a combination of measures in four categories, financial performance, customer knowledge, internal business processes, and learning and growth, to align individual, organizational and cross-departmental initiatives. They expect that the BSC will help companies test and update their strategy and meet their customer's needs and shareholder's objectives more effectively. Measuring the knowledge assets promotes strategic organizational learning (Brooking, 1996; Edvinsson, 1997; Massey et al., 2001; Roos and Roos, 1998; Sakaiya, 1991; Stewart, 1997; Teece, 1998, 2000), and generates the renewable organizational capabilities required to meet customer expectations on an ongoing basis (Leonard and Sensiper, 1998).
Fourth category of KM studies are addressing a role of IT in KM. The role of IT is investigated in KM in general (Alavi and Leidner, 1999; Newell and Scarbrough, 1999; Pérez-Bustamante, 1999) or for specific KM activities in particular (e.g., Liou and Nunamaker, 1993; Khalifa, 1998; Fischer et al. 1999; Petraglia and Glass, 1999; Squires, 1999; Suthers, 1999). KMS, Knowledge Management System, is a specialized information system for KM using modern technologies (e.g. the Internet, intranets, browsers, data warehouses, and software agents) in order to systematize, facilitate, and expedite firm-wide KM (Alavi and Leidner, 1999; Rouse et al., 1998). The KMS researches consist primarily of general and conceptual principles of KMS (Davenport et al., 1996) and case studies of such systems in a handful of leading organizations (Alavi, 1997; Baird et al., 1997; Bartlett, 1996; Henderson and Sussman, 1997; Sensiper, 1997; Watts et al., 1997). Especially, Gray (2000) describes how KMS can enhance the effectiveness of teams that analyze complex, non-recurring problems by improving the way that team composition evolves. Knowledge mining is similar to data mining. However, Rouse et al. (1998) uses knowledge mining to extract some knowledge from several data sources and apply it for more complicated and value-added problems. Holsapple and Joshi (2001) argue that DSS could be used to get the right knowledge in the right form to the right persons at the right time. Several papers are tackled technically from the perspective of a strategic use of the Internet for KM activities. Dieng (2000) discusses the potential of the Internet and intranets in developing distributed KMS. XML-based meta language is developed for knowledge retrieval from the knowledge repository administered on the web (Martin and Eklund, 2000). A KMS prototype named PlanetOnto, operating on the Internet, is suggested to support an academic community to collaboratively construct and share an archive of news items (Domingue and Motta, 2000). Schwartz and Te'eni (2000) exploit the Internet and e-mail to disseminate knowledge. Rabarijaona et al. (2000) addresses using XML to support users to translate a corporate ontology into an annotation document type definition. A representational infrastructure and a computational DSS framework are suggested for creating design repositories on the Internet (Szykman et al., 2000) and assisting a distributed team of designers in conceptual design evaluation on the web (Caldwell et al., 2000).
Fifth category is to explore success and failure factors of KM. Davenport et al. (1996), from successful KM projects, find out eight key factors to help a company create, share, and use knowledge efficiently. Ruggles (1998) identifies the KM concepts from over 400 firms in U.S and Europe and finds out what the barriers to KM are. Success factors for KM include compensation against knowledge provider, incentive systems, organization culture, etc. (Glasser, 1999; Krogh, 1998).
From the literature survey about KM, we can conclude that there is no study denoting a research objective similar to this paper, which is to propose a new measurement metric named KMPI for assessing KM performance. Then the next step should be what methodology is adopted to accomplish our research objective.

3. METHODOLOGY

3.1 Fundamentals of KMPI and Research Hypotheses

KCP includes a whole process of knowledge diffusion within a firm- knowledge creation, knowledge accumulation, knowledge sharing, knowledge utilization, knowledge internalization. KCP has a dynamic nature because it represents a knowledge flow concept where five components of knowledge circulation are interlinked with each other. Since KCP denotes a knowledge flow concept, and it is dynamic with time, we introduce time *t* into the KMPI function.

The effectiveness of KCP is influenced by various facets of organization culture like human relationships, degree of harmony between decision-making entities, quality of work process, strategic alliance with vendors, customers' trust, effectiveness of strategic management, and CEO's character and vision, etc., all of which in turn influence the management performance. Therefore, the proposed KMPI can be used to represent the KM performance. We assume that KCP has always been continuing since the firms started, and that organizational knowledge increases as KCP supports management activities from the knowledge-based perspective.

Tuomi (2000) suggests a reversed hierarchy of knowledge in which organizational knowledge is created after knowledge from which information is given meaning, and data emerge as a byproduct of cognitive artifacts. The proposed KMPI increases only if the KCP efficiency is improved, which is theoretically supported by Tuomi (2000)'s argument in that the existence of knowledge can create a form of competence and organizational knowledge, and management performance may be enhanced.

At this point, let us investigate the five components of KCP to clearly understand why the proposed KMPI can represent the quality of organizational knowledge, and the firm's management performance. The first component of KCP is knowledge creation which is concerned with creating a variety of knowledge, tacit or explicit. Knowledge creation is accelerated by massive synergistic interrelations of a lot of individuals having diverse backgrounds. Knowledge accumulation is a second knowledge flow component of KCP, which is stored into a knowledge repository. All the individuals in firms can have access to it to get relevant knowledge for their works or decision problems. Especially, the knowledge accumulated in firms for years can play an important role in eliminating various obstacles and inefficiencies and improving management performance, which is then called organizational knowledge (Walsh and Ungson, 1991). However, if knowledge created through management activities for years is not accumulated systematically either in electronically deliverable formats (O'Leary, 1988abc) or in structured documents, it cannot be used usefully for future decision-making needs. In this respect, knowledge created in various reasons and ways and forms should be accumulated in a form of organizational memory information systems (Stein and Zwass, 1995). Third component of KCP is knowledge sharing which promotes diffusion of knowledge in firms, and also contributes to making work process an intelligent and knowledge-intensive. In this situation, workers feel themselves so called *knowledge worker* (Sviokla, 1996). If knowledge workers can find knowledge necessary for processing their works successfully from the knowledge source administered by firms, then they are able to easily apply it to complete such works successfully. The knowledge-intensive work process requires integration of multiple knowledge much more for obtaining improved performance (Davenport et al., 1996). Knowledge utilization, fourth component of KCP, may be observed and performed in all the levels of management activities in firms. As the work processes need to become more knowledge-intensive in the aftermath of applying KM, one of the popular forms of knowledge utilization is to adopt best practice from other leading firms and find some knowledge relevant to us and apply it (O'Dell and Grayson, 1998). The fifth component of KCP is knowledge internalization which may occur when individual workers find some knowledge relevant to their works, and apply and obtain what they expect. Therefore, knowledge internalization may give rise to another knowledge, either new or modified one. In this way, knowledge internalization is providing a basis for more active knowledge creation. Nonaka and Konno (1998) suggest a concept of *Ba* where knowledge can be internalized more easily and created after all.

Based on the arguments above, the whole process of KCP is cycling from knowledge creation to knowledge accumulation to knowledge sharing to knowledge utilization to knowledge internalization. Knowledge accumulated in firms is a byproduct of KCP. Therefore, KCP has a concept of flow, and speed. If the flowing speed of KCP is fast, then we may assume that knowledge, its byproduct, is accumulated, shared, utilized, and internalized as fast, and that management performance increases, and that the proposed KMPI will improve after all.

In this way, KCP has an influence on the efficiency of work processes, and management activities performance. Based on the argument about KCP characteristics above, we claim that KMPI, which is assumed to be heavily influenced by KCP, can measure the quality of organizational knowledge, and that it is related directly and/or indirectly with firms' management performance. Therefore, we hypothesize that such firms with organizational knowledge of a good quality, will increase KMPI, and that those firms with greater KMPI will represent improved management performance. We adopt three specific measures like stock price, PER, R&D expenditure to translate management performance into tangible statistics. Then we can posit the following three research hypotheses.

Hypothesis 1: If KMPI is greater, then stock price is significantly better.

Hypothesis 2: If KMPI is greater, then PER is significantly better.

Hypothesis 3: If KMPI is greater, then R&D expenditure is significantly better.

3.2 KMPI Function

As knowledge beneficial to making work processes knowledge-intensive and improving management performance is accumulated in organization, then organization memory quality will increase with time, causing KMPI to increase gradually with an upper limit. The increase in KMPI per unit time is small at first, then increases rapidly and finally slows down. This rational can be described as follows. As workers learn to get accustomed to KCP - creating, accumulating, sharing, utilizing, and internalizing knowledge- in processing their works and integrate it with existing operations, the rate at which KMPI increase is small. The rate then increases as workers become familiar with applying KCP to their work processes. However, the rate slows down as KMPI approaches the limit of what can be gained from applying KCP to works. Stated formally, the impact of KCP application at time *t* is proportional to the KMPI gained at time *t-1* (i.e., $KMPI_{t-1}$) relative to the maximum possible KMPI gains from the KCP application (i.e., 1) and the remaining KMPI yet to be gained (i.e., $1-KMPI_{t-1}$). This description of KMPI over time *t* can be expressed as

$$\frac{dKMPI}{dt} = -KCP(1 - KMPI_{t-1}) \quad \text{(Eq 1)}$$

where KCP indicates a term denoting efficiency of KM in organization, which can be described as a function of five knowledge circulation processes. Solving (Eq 1) for KMPI yields

$$KMPI_t = \frac{1}{1 + e^{a + KCP \cdot t}} \quad \text{(Eq 2)}$$

Equation 2 is the S-shaped logistic model, where 1 is the upper bound on the KMPI from the KCP application, while *a* and *KCP* determine the shape of the curve. We assume that constant a is zero because each organization is supposed to start with very small KMPI. Then next step for calculating KMPI is to compute KCP which will be described empirically in the next section. Therefore, final equation form for KMPI is as follows.

$$KMPI_t = \frac{1}{1 + e^{KCP \cdot t}} \quad \text{(Eq 3)}$$

As noted previously, we suppose that KCP term in (Eq 3) is determined by five knowledge circulation processes. Stated empirically, KCP term in (Eq 3) is a function of *relative weight of eigenvalue* (RWE) of each knowledge circulation component multiplied by *average factor value* (AFV) of the corresponding knowledge circulation component.

$$\begin{aligned} KCP = {} & RWE_{KC} \cdot AFV_{KC} + RWE_{KA} \cdot AFV_{KA} \\ & + RWE_{KS} \cdot AFV_{KS} + RWE_{KU} \cdot AFV_{KU} + RWE_{KI} \cdot AFV_{KI} \end{aligned} \quad \text{(Eq 4)}$$

where KC means knowledge creation, KA knowledge accumulation, KS knowledge sharing, KU knowledge utilization, and KI knowledge internalization. How to compute RWE and AFV will be described in the next section from an empirical perspective.

4. THE EMPIRICAL STUDY

4.1 Survey Instrument Development

The process of designing the survey is notably influenced by Churchill (1979)'s recommendations for developing reliable and valid measures. Initially, questionnaire with 40 questions was prepared relating to five components of KCP. Open-ended interviews were used in the initial stages of instrument development. Two professors and four doctoral candidates and two practitioners, all of whom have been studying or practicing KM for years, were interviewed to ensure the questionnaire variables' face validity. Discussions with two KM professors on each variable helped in developing operational measures. Upon completion of the interviews, a pretest was conducted in which 18 executives from 18 companies were asked individually to evaluate the instrument and comment on the clarity of instructions and understandability of individual items. All of them responded, and based on the feedback received, 7 items were deleted from original 40 items. We concluded after evaluation that questionnaire using a seven-point scale, ranging from 1: "strongly disagree" to 4: "neutral" to 7: "strongly agree", is appropriate for measuring KMPI as intended in research design.

4.2 Data Collection

A cross-sectional field survey of companies in KOSDAQ market in Korea was conducted. A directory of companies compiled by a securities brokerage firm operating in a stock market was used as the sampling frame. This directory consists of organizations that at least one of the following three criteria:
(1) They were members of the KOSDAQ market.
(2) Their operating years are similar to each other because our definition of KM, causing KCP application to start with the foundation of company and then KMPI to increase gradually from company inauguration date, requires the almost same years of operation to avoid biases in measuring KMPI. Since KOSDAQ market opened in 1996, those companies surveyed have almost same five years of operation.
(3) They reported annual financial reports officially in line with formal official accounting standards imposed by KOSDAQ market.
A senior executive of each organization surveyed was asked to respond to questionnaire. While using a single source from each organization has its limitations, these are overcome to some degree by identifying the senior executive as executives most "informed" about KM and KCP and associated variables within each organization. A similar use of the "key informant" approach has been suggested for survey research and has been adopted by several IS researchers (Sethi and King, 1991). Surveys were sent to senior executives in 250 randomly selected organizations which met three criteria above. 101 usable responses were received, providing a response rate of 40.4 percent.

4.3 Sample Description

Table 3 provides a profile of the respondents by the number of full-time employees and sales volume. All sizes are well represented in our study example.

Table 3. Distribution of Respondents

(a) Distribution by Sales volume

Sales Volume (Unit: $1,000)	No. of Respondents	Percentage
$1,000 ≤	16	15.9
$1,000 – $10,000	28	27.7
$10,000 - $100,000	42	41.5
≥ $100,000	15	14.9
Total	101	100%

(b) Distribution by Full-time Employees Size

No. of full-time employees	No. of Respondents	Percentage
20 ≤	18	17.8
20 – 50	37	36.6
50 – 100	35	34.7
≥ 100	11	10.9
Total	101	100%

Table 4. Factor Structure of variables (N=101)

Factor	Eigenvalue	Cronbach's Alpha	Items	Factor loadings	Convergent validity
Knowledge Utilization	4.1307	0.86	There exist research and education programs	0.8002	0.86
			Team work is promoted by utilizing organization-wide information and knowledge	0.6437	0.68
			EDI is extensively used to facilitate processing tasks	0.6179	0.72
			There exist incentive and benefit policies for new ideas suggestion through utilizing existing knowledge	0.5327	0.67
			There exists a culture encouraging knowledge sharing	0.5199	0.71
			Work flow diagrams are required and used for performing tasks	0.5095	0.68
Knowledge Accumulation	4.1092	0.83	We refer to corporate database before processing tasks	0.7164	0.62
			We try to store know-how about new tasks design and development	0.6817	0.65
			We try to store legal guidelines and policies related to tasks	0.6729	0.69
			We extensively search through customer database and task-related database to obtain knowledge necessary for tasks	0.5687	0.66
			We document such knowledge needed for tasks	0.5524	0.81
			We summarize education results and store them	0.5400	0.65
			We are able to administer knowledge necessary for tasks systematically and store it for further usage	0.5081	0.85
Knowledge Internalization from-1	3.2388	0.77	I have a unique know-how for tasks	0.7134	0.72
			Professional knowledge such as customer knowledge and demand forecasting is managed systematically	0.6444	0.60
			Organization-wide standards for information resources are built	0.6211	0.71
			Employees are given education opportunity to improve adaptability to new tasks	0.5957	0.66
			University-administered education is offered to enhance employees' ability to perform tasks	0.5695	0.75
			Organization-wide knowledge and information are updated regularly and maintained well	0.5036	0.70
Knowledge Internalization from -2	2.4825	0.78	I can learn knowledge necessary for new tasks	0.6997	0.63
			I can refer to best practices and apply them to my tasks	0.6527	0.62
			I can use Internet to obtain knowledge for tasks	0.5633	0.69
Knowledge Sharing	2.3504	0.75	We share information and knowledge necessary for tasks	0.8760	0.64
			We improve task efficiency by sharing information and knowledge	0.7751	0.73
			We developed information systems like intranet and electronic bulletin board to share information and knowledge	0.7178	0.71
			We promote sharing necessary information and knowledge with other teams	0.5422	0.61
Knowledge Creation-1	2.3379	0.72	I often use an electronic bulletin board to analyze tasks	0.6434	0.62
			Predecessor gave me detailed introduction on my tasks	0.6246	0.64
			I fully understand core knowledge necessary for my tasks	0.5521	0.66
Knowledge Creation-2	2.0096	0.70	I obtain useful information and suggestions through idea-brainstorming meeting without spending too much time	0.7505	0.63
			I search information for tasks from various knowledge sources administered by organization	0.5628	0.67
			I understand computer programs needed to perform tasks and use them well	0.5482	0.64
			I am ready to accept new knowledge and apply it to my tasks when necessary	0.5321	0.71

4.4 Measures

4.3.1 Knowledge Creation

To measure knowledge creation, two constructs were operationalized-tasks understandings and information understandings. Tasks understandings are measured by three items (Nonaka and Takeguchi, 1995; Tuomi, 2000)– (1) I often use an electronic bulletin board to analyze tasks, (2) Predecessor gave me detailed introduction on my tasks, (3) I fully understand core knowledge necessary for my tasks. Information understandings are measured by four items (Leonard and Sensiper, 1998; Saint-Onge, 1998)– (4) I obtain useful information and suggestions through idea-brainstorming meeting without spending too much time, (5) I am ready to

Table 5. Average Factor Value

Organization	KC	KA	KS	KU	KI
com1	0.391	-0.679	-0.298	-0.312	0.350
com2	-0.781	0.197	0.554	0.477	1.057
com3	0.025	0.097	1.322	-1.707	-1.699
com4	2.383	1.498	0.473	1.848	0.590
com5	0.967	-0.616	-0.601	-1.111	0.378
com6	-0.282	-0.771	0.675	-0.650	-0.266
com7	0.784	-0.093	-0.348	-0.676	-0.706
com8	0.194	-1.387	0.280	0.259	0.010
com9	0.314	0.423	-1.255	-1.036	-1.151
com10	-0.854	0.441	0.377	1.067	-0.314
com11	0.497	-1.558	-1.433	-0.892	-0.416
com12	0.714	0.486	0.928	1.255	-1.664
com13	-1.061	-0.205	0.058	0.585	0.612
com14	0.626	-0.707	0.384	-0.328	0.342
com15	1.783	-2.509	-0.515	1.414	-1.225
com16	-0.175	-0.187	0.049	-1.550	0.005
com17	-1.975	-0.442	-0.554	2.191	1.447
com18	1.214	0.783	-0.027	-0.565	-0.935
com19	0.376	0.644	-0.560	0.171	-0.089
com20	1.884	-0.754	0.107	0.010	0.071
com21	0.188	0.520	0.200	-1.790	-0.272
com22	1.362	0.176	-0.261	-0.035	-0.235
com23	0.436	0.217	-0.140	-0.004	1.086
com24	-0.762	1.295	-0.483	-0.658	-0.539
com25	2.790	1.023	1.026	1.564	0.550
com26	-0.239	-0.160	-0.002	-0.572	0.190
com27	0.936	-0.331	-0.156	0.772	-0.115
com28	0.576	0.138	1.032	0.700	-0.970
com29	-0.219	1.522	-0.184	1.400	0.568
com30	0.421	-0.242	0.274	-1.683	-0.668
com31	-0.227	0.338	-0.095	-0.972	0.612
com32	0.281	0.383	-0.865	0.071	0.421
com33	-0.537	-0.030	1.399	0.236	-0.604
com34	-1.231	0.292	-1.051	0.006	-0.163
com35	-0.596	0.408	-1.475	0.479	1.008
com36	0.473	-0.451	-0.222	-1.954	-0.049
com37	-0.242	0.445	1.372	0.928	0.660
com38	-0.024	-1.048	0.696	0.308	-1.125
com39	-1.018	-0.071	-0.508	0.734	0.944
com40	-0.279	-0.153	0.574	-0.443	0.017
com41	-0.906	-0.912	0.516	0.279	0.373
com42	-0.041	-0.145	-0.811	-0.016	-0.376
com43	1.925	-0.061	0.492	1.539	0.538
com44	0.227	0.759	-1.805	-0.174	-0.236
com45	-0.219	-0.651	-0.117	-0.200	0.137
com46	-0.040	0.143	-0.373	-0.842	0.251
com47	-0.360	0.905	0.384	0.661	-0.081
com48	1.229	-0.444	1.324	-1.886	-1.024
com49	0.644	0.644	-0.012	0.278	0.020
com50	-0.757	0.087	0.434	0.244	-0.878
com51	0.202	-1.643	-0.344	0.155	-0.416
com52	0.936	0.263	0.320	1.159	0.239
com53	-1.146	-0.486	-0.858	0.028	-0.050
com54	0.098	0.847	0.197	0.319	-0.380
com55	0.238	0.858	-0.578	1.054	0.270
com56	0.601	0.767	0.396	0.683	0.430
com57	-0.855	0.783	1.277	-0.612	0.218
com58	-0.601	0.485	-0.181	0.219	-0.162
com59	0.723	-0.579	-0.481	0.684	-0.560
com60	-0.526	-1.552	0.146	-0.390	-0.578
com61	-0.765	-0.337	0.152	-0.529	0.879
com62	-1.154	-0.057	0.484	0.331	0.021
com63	1.004	1.000	0.504	1.113	-0.319
com64	2.271	0.370	-0.258	3.554	-0.484
com65	-1.412	0.650	0.332	0.512	-0.930
com66	0.971	0.808	-0.972	0.796	0.523
com67	-0.943	-0.006	0.099	-0.790	0.299
com68	-1.161	-0.439	0.821	-0.167	-0.156
com69	-0.933	-0.316	1.312	-1.421	1.028
com70	0.813	-0.238	1.197	-0.856	0.077
com71	0.627	0.440	0.055	1.067	0.994
com72	-0.516	0.070	-0.049	0.187	-1.063
com73	0.715	0.713	0.224	0.083	0.190
com74	1.601	-0.639	-0.466	-0.635	-0.588
com75	-2.969	0.367	0.810	-1.289	-1.195
com76	0.079	0.400	0.249	0.050	-0.220
com77	0.752	-0.721	0.912	1.052	0.906
com78	0.736	0.572	0.298	0.451	0.572
com79	0.590	0.264	-0.682	0.147	0.994
com80	0.637	0.734	0.069	-0.583	1.068
com81	-1.743	-0.114	-0.875	-1.391	-0.759
com82	0.627	0.440	0.055	1.067	0.994
com83	-2.969	0.367	0.810	-1.289	-1.195
com84	0.079	0.400	0.249	0.050	-0.220
com85	0.590	0.264	-0.682	0.147	0.994
com86	0.637	0.734	0.069	-0.583	1.068
com87	0.644	0.644	-0.012	0.278	0.020
com88	0.238	0.858	-0.578	1.054	0.270
com89	-0.526	1.614	0.146	-0.390	0.564
com90	0.436	0.217	-0.140	-0.004	1.086
com91	0.936	-0.331	-0.156	0.772	-0.115
com92	-0.227	0.338	-0.095	-0.972	0.612
com93	-0.282	-0.771	0.675	-0.650	-0.266
com94	0.714	0.486	0.928	1.255	-1.664
com95	1.214	0.783	-0.027	-0.565	-0.935
com96	-0.041	-0.145	-0.811	-0.016	-0.376
com97	1.925	-0.061	0.492	1.539	0.538
com98	-0.297	0.759	-1.805	-0.174	0.288
com99	-1.146	-0.486	-0.858	0.028	-0.050
com100	0.098	0.847	0.197	0.319	-0.380
com101	-0.540	-1.048	0.696	0.308	-1.125

accept new knowledge and apply it to my tasks when necessary, (6) I understand computer programs needed to perform tasks and use them well, (7) I search information for tasks from various knowledge sources administered by organization.

Table 6. Relative Weight of Eigenvalue (RWE)

Factor	Eigenvalue	RWE
Knowledge Creation	4.348	0.211
Knowledge Accumulation	4.110	0.199
Knowledge Sharing	2.350	0.114
Knowledge Utilization	4.131	0.200
Knowledge Internalization	5.722	0.276
Total	20.661	1

4.3.2 Knowledge Accumulation

An instrument knowledge accumulation was tested by three constructs – database utilization, systematic management of task knowledge, and individual capacity for accumulation. Database utilization was operationalized by two items (O'Leary, 1998abc; Tuomi, 2000)- (1) We refer to corporate database before processing tasks, (2) We extensively search through customer database and task-related database to obtain knowledge necessary for tasks. Systematic management of task knowledge was operationalized by three items- (3) We try to store know-how about new tasks design and development, (4) We try to store legal guidelines and policies related to tasks, (5) We are able to administer knowledge necessary for tasks systematically and store it for further usage. Individual capacity for accumulation was operationalized by two items- (6) We document such knowledge needed for tasks, (7) We summarize education results and store them.

4.3.3 Knowledge Sharing

Degree of sharing knowledge is dependent upon constructs such as core knowledge sharing and knowledge sharing in organization. Core knowledge sharing was measured by two items (Lank, 1997; Sviokla, 1996)- (1) We share information and knowledge necessary for tasks, (2) We improve task efficiency by sharing information and knowledge. Knowledge sharing in organization was operationalized by two items (Davenport et al., 1996; Ruggles, 1998)- (3) We promote sharing necessary information and knowledge with other teams, (4) We developed information systems like intranet and electronic bulletin board to share information and knowledge.

4.3.4 Knowledge Utilization

Knowledge utilization depends on two constructs- degree of knowledge utilization in organization, and knowledge utilization culture. The former was operationalized by three items (O'Dell and Grayson, 1998; Weber et al., 1990; Blanning and Daivd, 1995)- (1) Team work is promoted by utilizing organization-wide information and knowledge, (2) EDI is extensively used to facilitate processing tasks, (3) Work flow diagrams are required and used for performing tasks. The latter was operationalized by three items (Leonard and Sensiper, 1998; Wiseman, 1988)- (4) There exists a culture encouraging knowledge sharing, (5) There exist incentive and benefit policies for new ideas suggestion through utilizing existing knowledge, (6) There exist research and education programs.

4.3.5 Knowledge Internalization

Knowledge internalization was measured by three constructs- capability to internalize task-related knowledge, education opportunity, level of organization learning. Capability to internalize task-related knowledge was operationalized by four

Table 7. KMPI Calculation

Organization	KMPI	Organization	KMPI	Organization	KMPI	Organization	KMPI
com4	0.800	com49	0.580	com95	0.509	com24	0.441
com25	0.798	com87	0.580	com13	0.507	com67	0.434
com64	0.750	com89	0.576	com31	0.496	com21	0.434
com43	0.712	com20	0.570	com92	0.496	com65	0.429
com97	0.712	com17	0.567	com59	0.491	com50	0.429
com71	0.671	com47	0.564	com58	0.487	com68	0.422
com82	0.671	com27	0.559	com1	0.487	com48	0.416
com29	0.662	com91	0.559	com98	0.482	com6	0.416
com63	0.646	com22	0.555	com61	0.481	com93	0.416
com52	0.642	com54	0.543	com33	0.480	com72	0.412
com56	0.641	com100	0.543	com5	0.474	com34	0.410
com77	0.641	com32	0.542	com44	0.473	com16	0.407
com37	0.637	com35	0.540	com40	0.473	com38	0.405
com66	0.636	com19	0.538	com69	0.472	com51	0.400
com78	0.634	com12	0.536	com46	0.470	com36	0.397
com80	0.615	com94	0.536	com62	0.468	com53	0.391
com86	0.615	com28	0.535	com74	0.467	com99	0.391
com55	0.608	com39	0.530	com26	0.464	com30	0.390
com88	0.608	com70	0.528	com8	0.463	com101	0.379
com23	0.603	com10	0.519	com41	0.461	com9	0.373
com90	0.603	com76	0.518	com45	0.452	com3	0.346
com79	0.599	com84	0.518	com7	0.445	com60	0.345
com85	0.599	com14	0.516	com42	0.441	com11	0.340
com73	0.596	com57	0.515	com96	0.441	com81	0.273
com2	0.580	com18	0.509	com15	0.441	com75	0.259

items- (1) I have a unique know-how for tasks, (2) I can learn knowledge necessary for new tasks, (3) I can use Internet to obtain knowledge for tasks, (4) I can refer to best practices and apply them to my tasks. Education opportunity was operationalized by two items- (5) Employees are given education opportunity to improve adaptability to new tasks, (6) University-administered education is offered to enhance employees' ability to perform tasks. Level of organization learning was operationalized by three items- (7) Professional knowledge such as customer knowledge and demand forecasting is managed systematically, (8) Organization-wide standards for information resources are built, (9) Organization-wide knowledge and information are updated regularly and maintained well.

4.5 Data Analyses Procedure

Preliminary factor analysis of items in each of the constructs validated the measures that were later used in the KMPI calculation model (Eq 3) and (Eq 4). Exploratory factor analysis was adopted with orthogonal rotation method (Hair et al., 1998). Seven factors were found with cronbach alpha value being greater than 0.7, which indicates that internal consistency is guaranteed in each factor dimension. Table 4 shows factor structure of variables, where reliability and convergent validity are significant because cronbach's alpha is greater than or equal to 0.70, and all convergent validity is greater than 0.60 (Hair et al., 1998). Table 5 and 6 summarize RWE and AFV, all of which are required to calculate KMPI shown in Table 7. Table 8 shows correlation test between KMPI and three financial measures. Hypotheses 1 and 2 are proved with 0.1 significance level, while hypothesis 3 is proved with 0.05 significance level.

The empirical results in Tables 4 through 7 show that as theorized, those five components of KCP affect KMPI significantly, which in turn represents the quality of organization memory that is utilized in a wide variety of decision-makings in an organization. If the quality of organization memory is good, then we can easily conjecture that management performance improves significantly.

Table 8. Correlation between KMPI and Three Financial Measures

Financial Measures	Correlation with KMPI
Stock Price	0.233*
PER	0.213*
R&D Expenditure	0.259**

*: $p<0.1$, **: $p<0.05$

5. CONCLUDING REMARKS

In this paper, we have presented a discussion of a close relationship between KMPI and KCP. Our study shows that there is no tension between the effects of KCP and KMPI. As the efficiency of five components of KCP increases, then KMPI becomes greater through the logistic model. Those five components of KCP are knowledge creation, knowledge accumulation, knowledge sharing, knowledge utilization, and knowledge internalization. Based on a review of a broad range of relevant literature, several conclusions may be drawn from our study.

(1) Extensive literature review revealed the complexity and multi-faceted nature of organizational knowledge and KM, and the need for developing a new measurement metric to assess the KM performance. To deal with complex nature of organizational knowledge and its contribution to KM performance, we introduced a concept of KCP and applied it to devise a function of KMPI.

(2) KMPI function is basically a logistic model in which the contribution of organizational knowledge accumulated by performing KM for years starts with a slow growth rate and increases fast and slows down at some point in time to a mature level.

(3) Power of KMPI to represent financial performance of firms was tested statistically. We used three major financial indices such as stock price, PER, and R&D expenditure. We proved that correlation between KMPI and those three indices is statistically significant.

IT has a strong impact on the effectives of five components of KCP. Especially, the Internet may become a crucial factor for making KMPI successful because the Internet use in daily management activities renders normal and essential. Based on this prospect, it is necessary to investigate the potential contribution of the Internet and consider it in improving the KMPI. We hope that this study may trigger future researches in this challenging field of evaluating the KM performance.

ACKNOWLEDGMENT

This study was supported by Korea Research Foundation Grant. (KRF-99-005-C00002)

REFERENCES

Alavi, M. and Leidner, D. (1999). Knowledge Management Systems: Emerging Views and Practices from the Field. *Proceedings of the 32nd Hawaii International Conference on System Science*s, CD-Rom Version

Alavi, M. (1997). KPMG Peat Marwick U.S.: One Giant Brain. Harvard Business School, Case 9-397-108

Alavi, M. and Leidner, D.E. (2001) "Review: knowledge management and knowledge management systems: conceptual foundations and research issues", *MIS Quarterly*, 25 (1), pp. 107-136.

Baird, L., J. Henderson, and Watts, S. (1997). "Learning Networks: An Analysis of the Center for Army Lessons Learned," *Human resource Management Journal*, 7(2).

Barney, J.B. (1991). "Firm Resources and Sustained Competitive Advantage", *Journal of Management*, 17, pp.99-120.

Bartlett, C. (1996). "McKinsey & Company: Managing Knowledge and Learning," Harvard Business School Case 9-396-357.

Blanning, R.W. and K. David, (1995). *Organizations Intelligence,* IEEE Computer Society Press, pp.39-50.

Brooking, A. (1996). *Intellectual Capital: Core Asset for the Third Millennium Enterprise.* London: International Thomson

Business Press.

Caldwell, N.H.M., Rodgers. P.A., Huxor, A.P. and Clarkson, P.J. (2000). “Web-Based Knowledge Management for Distributed Design”, *IEEE Intelligent Systems & Their Applications*, 15 (3), pp.40-47.

Carlsson, S.A., El Sawy, O.A., Eriksson, I., and Raven, A. (1996). " Gaining Competitive Advantage Through Shared Knowledge Creation: In Search of a New Design Theory for Strategic Information Systems," *in Proceedings of the Fourth European Conference on Information Systems*, J. Dias Coelho, T. Jelassi, W. Konig, H. Krcmar, R. O'callaghan, and M. Saaksjarvi(eds.), Lisbon.

Choo, C. W. (1998), *The knowing organization : how organizations use information to construct meaning, create knowledge, and make decisions*, New York : Oxford University Press

Churchill, G.A. (1979). “A paradigm for development better measures of marketing constructs”, *Journal of marketing Research*, 16, pp.64-73.

Cole, R.E. (1998). “Introduction:, *California Management Review*, 45(3), pp.15-21.

Davenport, T.D., Jarvenpaa, S.L. and Beers, M.C. (1996). “Improving Knowledge work Processes", *Sloan Management Review*, 37(4), pp.53-65.

Davenport, T.H. and Grover, V., (2001). “General perspectives on knowledge management: fostering a research agenda”, *Journal of Management Information Systems*, 18 (1), pp. 5-22.

Dieng, R. (2000). “Guest Editor’s Introduction: Knowledge Management and the Internet”, *IEEE Intelligent Systems & Their Applications*, 15 (3), pp.14-17.

Domingue, J. and Motta, E. (2000). “PlanetOnto: From News Publishing to Integrated Knowledge Management Support”, *IEEE Intelligent Systems & Their Applications*, 15 (3), pp.26-32.

Edvinsson, L., (1997). "Developing intellectual capital at Skandia", *Long Range Planning, 30(3),* pp.366-373.

El Sawy, O.A., G.M. Gomes, and Gonzalez, M.V. (1986). Preserving institutional memory: the management of history as an organizational resource, *Academy of Management Best Paper Proceedings*, 37, pp.111-122

Elliott, S. (1997), Arthur Andersen Maximizes Its Core Commodity Through Comprehensive Knowledge Management, *Knowledge Management in Practic*e, Issue 9, August / September

Fischer, M., Brown, A., Attwell, G. and Owen, M. (1999), The Development of an Experience-Based Documentation System for Maintenance Workers in Germany, *Proceedings of the 32nd Hawaii International Conference on System Science*s, CD-Rom Version

Glasser, P. (1999), The Knowledge Factor, http://www.cio.com/archive/010199_know.html

Gold, A.H., Malhotra, A. and Segars, A.H. (2001). “Knowledge management: an organizational capabilities perspective”, *Journal of Management Information Systems*, 18 (1), pp. 185-214.

Grant, R. M. (1996a). "Prospering in Dynamically-Competitive Environments: Organizational Capability as Knowledge Integration," *Organization Science, 7(4),* pp. 375-387.

Grant, R. M. (1996b). "Toward a Knowledge-based Theory of the Firm," *Strategic Management* Journal(17), Winter Special Issue, pp. 109-122.

Gray, P.H. (2000). "The effects of knowledge management systems on emergent teams: towards a research model", *Journal of Strategic Information Systems*, 9, pp.175-191

Hair, J.F., Anderson, R.E., Tatham, R.L., and Black, W.C. (1998). *Multivariate Data Analysis*, 5th Edition Prentice Hall International Inc.

Henderson, J.C., and Sussman, S.W. (1997). "Creating and Exploiting Knowledge for Fast-Cycle Organizational Response: The Center for Army Lessons Learned." *Working Paper* No. 96-39, Boston University.

Holsapple, C.W. and Joshi, K.D. (1999), Description and Analysis of Existing Knowledge Management Frameworks, *Proceedings of the 32nd Hawaii International Conference on System Science*s, CD-Rom Version

Holsapple, C.W. and Joshi, K.D. (2001). “Organizational knowledge resources”, *Decision Support Systems*, 31, pp. 39–54.

Kaplan, R.S. and Norton, D.P. (1992), The Balanced Scorecard Measures that Drive Performance, *Harvard Business Revie*w, January-February, 70(1), pp. 71-80

Khalifa, M. (1998), Effects of Hypertext on Knowledge Construction, *Proceedings of the 31st Hawaii International Conference on System Science*s, CD-Rom Version

Krogh, G.V. (1998). “Care in knowledge Creation", *California Management Review,* 40(3), pp.133-153.

Lank, E. (1997). “Leveraging Invisible Asset: The Human Factor", *Long Range Planning*, 30(3), pp.406-412.

Leonard-Barton, D., (1995), *The Wellsprings of Knowledge*. Boston, Mass.: Harvard Business School Press.

Leonard, D. and Sensiper, S.(1998). “The Role of Tacit Knowledge in Group Innovation", *California Management Review*, 40(3), pp.112-132.

Liou, Y.I. and Nunamaker, J.F. (1993), An Investigation into Knowledge Acquisition Using a Group Decision Support

System, *Information & Management*, 24, pp.121-132

Markus, L.M. (2001) “Toward a theory of knowledge reuse: types of knowledge reuse situations and factors in reuse”, *Journal of Management Information Systems*, 18 (1), pp. 57-94.

Martin, P. and Eklund, P.W. (2000). “knowledge Retrieval and the World Wide Web”, *IEEE Intelligent Systems & Their Applications*, 15 (3), pp.18-25.

Massey, A.P, Montoya-Weiss, M.M., and Holcom, K. (2001) “Re-engineering the customer relationship: leveraging knowledge assets at IBM”, *Decision Support Systems,* 32(2), pp. 155-170

McGill, M.E. and Slocum, J.W. (1994). *The Smarter Organization: How to Build a Business That Learns and Adapts to Marketplace Needs*, John Wiley & Sons.

McQueen, R. (1998). "Four Views of Knowledge and Knowledge Management," *in Proceedings of the Fourth Americas Conference on Information Systems*, E. Hoadley and I. Benbasat (eds.), pp. 609-611.

Newell, S. and Scarbrough, H. (1999), Intranets and Knowledge Management: Complex Processes and Ironic Outcomes, *Proceedings of the 32nd Hawaii International Conference on System Sciences*, CD-Rom Version

Nonaka, I. and Takeguchi. H.(1995). *The Knowledge Creating Company.* Oxford University Press.,

Nonaka, I. and Konno, N. (1998) “The Concept of Ba": Building a Foundation for Knowledge Creation", *California Management Review*, 40(3), pp.31-54.

O'Dell, C. and Grayson, C.J. (1998). “If Only We knew What We know: Identification and Transfer of internal Best Practices", *California Management Review*, 40(3), pp.154-174.

O'Leary, D.E. (1998a), “Enterprise Knowledge Management", *IEEE Computer, March,* pp. 54-61.

O'Leary, D.E. (1998b) “Using AI in Knowledge Management: Knowledge Bases and Ontologies", *IEEE Intelligent Systems*, *May/June*, pp.34-39.

O'Leary, D.E., (1998c) “Knowledge Management Systems: Converting and Connecting", *IEEE Intelligent Systems, May/June*, pp. 30-33.

Penrose, E.T. (1959). *The Theory of the Growth of the Firm*, Wiley, New York.

Pentland B.T. (1995), Information Systems and Organizational Learning: The Social Epistemology of Organizational Knowledge Systems, *Accounting, Management & Information Technology*, 5(1), pp. 1-21

Pérez-Bustamante, G. (1999), Knowledge Management in Agile Innovative Organizations, *Journal of Knowledge Management*, 3(1) , pp.6-17

Petraglia, J. and Glass, B. (1999), Technological Support for Knowledge Negotiation, *Proceedings of the 32nd Hawaii International Conference on System Sciences*, CD-Rom Version

Petrash, G. (1996), Dow's Journey to a Knowledge Value Management Culture, *European Management Journal*, 14(4), pp.365-373

Purser, R., Pasmore, W., and Tenkasi, R. (1992). The influence of deliberations on learning in new product development teams. *Journal of Engineering and Technology Management,* 9, pp.1-28.

Rabarijaona, A., Dieng, R., Corby, O. and Ouaddari, R. (2000). “Building and Searching an XML-Based Corporate Memory”, *IEEE Intelligent Systems & Their Applications*, 15 (3), pp.56-63.

Roos, J. and Roos, G. (1998). "*Intellectual Capital: Navigating in the New Business Landscape*", New York Univ. Press.

Roth,G. and Senge, P. (1995). "From Theory to Practice: Research Territory, Processes and Structure at an Organizational Learning Center, *Journal of Change Management*, 9(1), 93-108.

Rouse, W.B., Thomas, B.S. and Boff, K.R. (1998). "Knowledge Maps for Knowledge Mining: Application to R&D/Technology Management", *IEEE Transactions On Systems, Man, And Cybernetics Part C: Applications And Reviews*, 28(3), pp.309-317.

Ruggles, R. (1998). “The State of the Notion: Knowledge Management in Practice", *California Management Review*, 40(3), pp.80-89.

Saint-Onge, H. (1998). “How Knowledge Management Adds Critical Value to Distribution Channel Management”, *Journal of Systemic Knowledge Management*, http://www.free-press.com

Sakaiya, T. (1991). *The Knowledge-Value Revolution, or A History of The Future*, Kodansha International, Tokyo.

Schatz, B.R. (1991-1992). Building an electronic community system, Journal of Management Information Systems, 8 (3), pp. 87-107.

Schubert, P., Lincke D., and Schmid, B. (1998). "A Global Knowledge Medium as a Virtual Community: The NetAcademy Concept," *in Proceedings of the Fourth Americas Conference on Information Systems*, E. Hoadley and I. Benbasat(eds.), Baltimore, MD, pp. 618-620.

Schwartz, D.G. and Te’eni, D. (2000). “Tying Knowledge to Action with kMail”, *IEEE Intelligent Systems & Their Applications*, 15 (3), pp.33-39.

Senge, P.M. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York, Doubleday/Currency.

Sensiper, S. (1997), “AMS Knowledge Centers,” Harvard Business School Case N9-697-068.

Sethi, V. and King, W.R. (1991). “Construct measurement in information system research: an illustration in strategic systems”, Decision Sciences, 22(3), pp.455-472.

Shaw, R.B. and Perkins, D.N.T. (1992). "Teaching organizations to learn: The power of productive failures." In Nadler, D. A., Gerstein, M. S., Shaw, R. B. *Organizational architecture: designs for changing organizations.* (pp. 175-208). San Francisco: Jossey-Bass.

Spender, J.C. (1996a). “Making Knowledge the Basis of a Dynamic Theory of the Firm”, *Strategic Management Journal*, 17, Special Issues, pp.45-62.

Spender, J.C. (1996b). “Organizational Knowledge, Learning, and Memory: Three Concepts in Search of a Theory”, *Journal of Organizational Change Management*, 9, pp.63-78.

Squires, D. (1999), Educational Software and Learning: Subversive Use and Volatile Design, *Proceedings of the 32nd Hawaii International Conference on System Science*s, CD-Rom Version

Stata, R. (1989). “Organizational learning: the key to management innovation", *Sloan Management Review*, 30(3), pp.63-74.

Stein, E.W. and V. Zwass (1995). Actualizing organizational memory with information systems. *Information Systems Research*, 6(2), pp. 86-117.

Stewart, T.A. (1997). *Human Capital: Intellectual Capital.* Doubleday/Currency, pp.79-106.

Suthers, D. (1999), Representational Support for Collaborative Inquiry, *Proceedings of the 32nd Hawaii International Conference on System Science*s, CD-Rom Version

Sveiby, K.E. (1998). *Intellectual capital: Thinking ahead.* Australian CPA.

Sviokla, J.J. (1996), “Knowledge Workers and Radically New Technology," *Sloan Management Review, 37(2)*, pp.25-40.

Szulanski, G. (1996), Exploring Internal Stickiness: Impediments to the Transfer of Best Practice within the Firm, *Strategic Management Journa*l, 17, Winter Special Issue, pp.27-43

Szykman, S., Sriram, R.D., Bochenek, C., Racz, J.W. and Senfaute, J. (2000). “Design Repositories: Engineering Design’s New Knowledge Base”, *IEEE Intelligent Systems & Their Applications*, 15 (3), pp.48-55.

Teece, D.J. (1998). “Capturing Value from Knowledge Asset: The New Economy, Markets for Know-how, and Intangible Assets”, *California Management Review*, 40(3), pp.55-79.

Teece, D.J., (2000). “Strategies for Managing Knowledge Assets: the Role of Firm Structure and Industrial Context”, *Long Range Planning*, 33(1), pp. 35-54.

Tuomi, I. (2000). "Data Is More Than Knowledge: Implications of the Reversed Knowledge Hierarchy for Knowledge Management and Organizational Memory", Journal of Management Information Systems, 16(3), pp.103- 118

Van de Ven, A. and D. Pooley (1992), “Learning while Innovating,” *Organizational Science*, 3, pp. 92-116.

Van der Spek R. and Spijkervet A. (1997), Knowledge Management: Dealing Intelligently with Knowledge, in Knowledge Management And Its Integrative Elements (eds. Liebowitz, J. and Wilcox, L.), New York: CRC Press

Walsh, J.P. and G.R. Ungson (1991). Organizational Memory, Academy of Management Journal, 16 (1), pp. 57-91.

Watson, R.T. (1999) *Data Management: Databases and Organizations* (2nd ed.), John Wiley, New York.

Watts, S.A., J.B. Thomas, and J.C. Henderson (1997), “Understanding ‘Strategic Learning’: Linking Organizational Learning, Sensemaking, and Knowledge Management,” *Academy of Management Meeting*, Boston MA.

Weber, E.S., Liou, Y.I., Chen, M., and J.F. Nunamaker (1990), “Toward More Intelligent Organizations," *Proceedings of the 23st Annual Hawaii International Conference on Systems Science*, *4*, pp.290-299.

Wernerfelt, B. (1984). “A Resources-Based View of the Firm”, *Strategic Management Journal*, 5, pp.171-180.

Wiseman, C. (1988). *Strategic Information Systems.* Homewood, IL: Irwin.

Zack M.H. (1998). "An Architecture for Managing Explicated Knowledge," *Sloan Management Review.*

Zack, M. H. (1999), Developing a Knowledge Strategy, *California Management Revie*w, 41(3), pp. 125-145

Enterprise Modelling: A Declarative Approach for *FBPML*

Yun-Heh Chen-Burger[1], **Austin Tate**[1], **Dave Robertson**[2]

[1] **AIAI, The University of Edinburgh**
80 South Bridge, Edinburgh, UK
email: jessicac@aiai.ed.ac.uk a.tate@ed.ac.uk

[2] **Centre for Intelligent Systems and their Applications**
The University of Edinburgh, UK
email: dr@dai.ed.ac.uk

Abstract. Enterprise Modelling (EM) methods are well-recognised for their value in describing complex, informal domains in an organised structure. EM methods are used in practice, particularly during the early stages of software system development, e.g. during the phase of business requirements elicitation. The built model, however, has not always provided direct input to software system development. Despite the provision of adequate training to understand and use EM methods, informality is often seen in enterprise models and presents a major obstacle. This paper focuses on one type of EM methods: business process modelling (BPM) methods. We advocate the use of a BPM language within a three-layer framework. The BPM language merges two main and complimentary business process representations, IDEF3 and PSL, to introduce a Fundamental Business Process Modelling Language (FBPML) that is designed for simplicity of use and under-pinned by rich formality that may be used directly to support software and workflow system development.

Key-words Business Process Modelling, IDEF3, PSL, Workflow Management, Business Modelling, BSDM, Formal Method, Enterprise Modelling, Collaborative (Web-based) Knowledge Management.

1 Introduction - The Gap

Enterprise modelling (EM) methods are well-recognised for their value in organising and describing a complex, informal domain in a more precise semi-formal structure that is intended for more objective understanding and analysis. Example EM methods are business modelling method, business modelling of IBM's BSDM (Business System Development Method) [13], process modelling method, IDEF0[18], IDEF3[17], PSL[21], RAD[19], RACD[3], CommonKADS Communication Model Language (CML)[26], organisational modelling, Ordit[7] Ulrich[10], capability modelling, [22] and (Enterprise) Ontology [23], [25], [9].

Despite their use, Enterprise Models have not always provided direct input for software system development. Obstacles include the necessary training required for users to learn conceptual modelling in general as well as the specific techniques required for the specific method applied. Generic knowledge acquisition techniques are also needed to elicit knowledge from the application domain. One other main obstacle is the lack of direct mapping from EM methods to software system development. Since EM methods are normally described at higher levels of abstraction which are independent of implementation issues, EM methods are often used merely as a description and analysis tool of the application domain. However, as EM methods often describe requirements from the business side, as opposed to from the technical side, the built Enterprise Models are natural candidates to provide a "blueprint" for business requirements when building software systems.

Figure 1 illustrates the gap that exists between Enterprise Models and common software systems built for organisations. It also proposes three possible means, all of them based on formal methods, *Quality Assurance, Mapping of Data Structure and Workflow System*, to bridge the gap by providing direct mappings between Enterprise Models and designing and building of software systems.

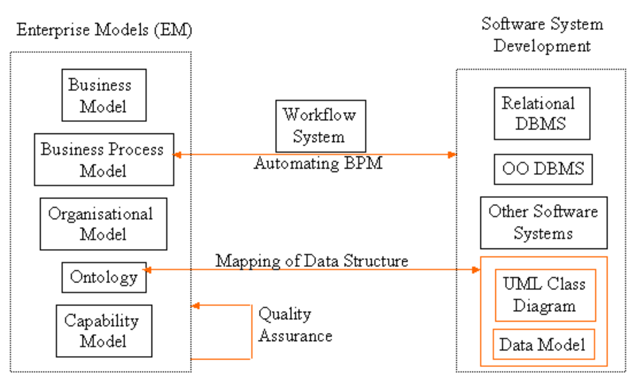


Figure 1. Bridging Gap between Enterprise Models and Software Systems

Formal methods may be used in various ways to facilitate communication between modellers and users of models, e.g. to make tacit information explicit and present it in different (maybe less technical and/or more familiar) forms, or to provide simulation functionalities to allow the reader to run through possible user scenarios in a state machine[2][20][11]. Automatic support such as knowledge sharing and inconsistency checking between different Enterprise Models, when a set of EM has been used, may also be done based on one commonly shared ontology [3]. The automatic support helps the modeller and user of the model understand a model in depth, therefore enhances their ability in error detection and model refinement. As a result, quality of the built models is improved. The refinement process based on computing support is indicated by the "Quality Assurance" arrow in the figure. Another way to bridge the gap is to provide a means to transfer data and knowledge that are held in the EM, particularly in an ontology, to software systems. This may be done by mapping an ontology to ER (Entity-Relational) Model (for Relational Databases) or to Class Diagram (for Object-Oriented Databases) or other types of data structures. This is indicated by the "Mapping of Data Structure" arrow.

This paper focuses on one type of EM method: Business Process Modelling (BPM) Method. One direct and obvious way to make use of BPM methods and to provide a direct input to software systems is to build a workflow system that is based on a business process model[8]. A definition of workflow, that is given by the Workflow Management Coalition, that describes its relationship with a business process is given below:

> *"The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."[8]*

Although the above approach seems obvious, in practice not all workflow systems have received the full benefit from business process modelling. The BPM approach towards building a workflow system is a recent and gradual approach over the past two years. This is different from the first generation workflow systems where BPM was not used[12]. The reasons for such phenomena are the lack of training and understanding of BPM methods and how they may be applied in an organisation. The business process when it is used often does not separate business and implementation logic, and hence, the resulting workflow system is not flexible in reaction to the dynamic and volatile environment within which the workflow system operates.

Last but not least, while BPM methods are normally described at a higher level of abstraction that enables flexibility for implementation, they do not provide sufficient details of additional information that must be included for process enactment. It is therefore beneficial to provide a means that maintains the flexibility of higher level descriptions, while at the same time providing sufficient information and a mechanism to carry out workflow[14].

This paper proposes a layered business process modelling approach that aims to lessen the above problems, therefore narrowing the gap. The paper also describes the design of FBPML (Fundamental Business Process Modelling Language) and how business processes based on it may be mapped to a visualisation of dynamic states of a workflow system in a collaborative enterprise environment.

2 An Ontology based Three-Layer BPM Framework

Figure 2 describes a layered business process modelling framework which provides the means to allow higher level business processes, objectives and policies to be carried forward and realised in the actual implementation of software (and manual) systems. The upper two levels of the framework describe business operations at a higher level of abstraction; the lower level of the framework describes how these business operations may be implemented in a software system. In this framework, design rationale of a software system is based on a company's objectives, hence the corresponding software system can be traced back to the initial business requirements and justified. Both of these enable the system to be coherent with the overall business aims.

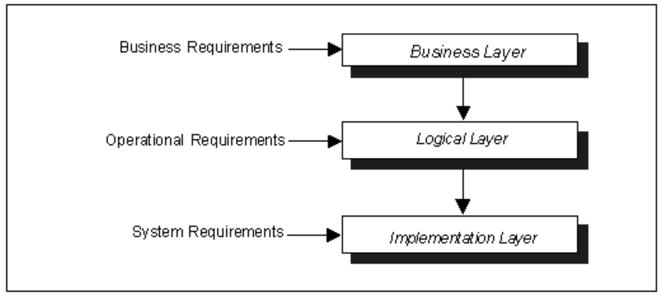


Figure 2. A Three-Layer Business Process Modelling Approach

The first layer, Business Layer, describes business requirements of an organisation, processes that are to be carried out by the organisation and information used by these processes. Information stored in this layer are higher level descriptions that may be written in informal or semi-formal documents. Examples are source data files, mission and organisation goal statements, business plans, and summary and vision of business operations. In this layer, information that is consolidated, such as business policies, longer lasting organisational structure and business-level decisions that are used as guidelines for developing business process models, is in general robust against change of technologies and (automated or manual) practices.

The second layer, Logical Layer, expresses a logical description of business processes. This description dictates the conditions and actions of business processes, the relationships between them as well as operational constraints on data that processes operate on. The Logical Layer is a (semi-formal) business process model that describes business operations in ordered activities. It extracts and formalises business requirements using computer understandable languages, while leaving the corresponding (informal) source data side by side in the model for reference and justification of its formal representation. It also interprets and elaborates the abstract requirements described in the Business Layer into more concrete constraints using the designed language to provide direct design guidelines for the implementation of the software system. The process modelling language, FBPML, that will be described in Section 4 resides in this layer.

The formality described in this layer allows automatic communication with the next layer, the Implementation Layer. Logical layer, however, does not consider the mechanism which may be used to enact the described processes. Such issues are dealt with in the Implementation Layer. Examples of such issues are the software paradigm deployed, software and hardware systems involved, integration issues, and programming languages used. Descriptions in the Logical Layer may have multiple mappings to descriptions in the Implementation Layer. This is particularly applicable in a complex or an agent architecture system where different components may have different functionalities and means to implement the same logical process. They also need to collaborate with each other to accomplish a business process.

The logical layer specifies all of the process-related and the core set of data-related integrity constraints so that the implemented system does not violate any business or operational constraint. Since a business process may be enacted by different system components and they may be carried out concurrently, the business process model provides a common and sharable knowledge base for process communication during enactment. Because a business process model captures operational logic and is independent of technologies used for implementation, it is more robust against changes of technologies.

The Implementation Layer gives detailed step-by-step algorithmic procedures for software modules that implement processes described in the Logical Layer. Such algorithmic procedures may be described in a process modelling language that is capable of describing implementation details, or languages similar to flow-control and data-flow diagrams, or other application or system specific languages. Implementation Layer tends to be technology-dependent, it may be changed very frequently. For instance, an introduction of a new user interface, software or hardware system component may or may not result in a change in the logical layer, but will probably cause a modification of the corresponding descriptions in the Implementation Layer. For this reason, processes given in the Implementation Layer are volatile and disposable, as new technologies become available. They may be easily changed without disturbing a business's operation in a principle way leaving the business a more flexible and agile system.

Information that is manipulated by logical processes is organised in a hierarchical fashion, i.e. a *Domain Ontology*. The Domain Ontology gives semantics of the information stored and is comparable to a subset of classes that may be used to store operation related information in a database. It includes common classes (or a part of the schema for a "relational system") that are shared by different logical processes to allow them to exchange information under a standardised business practice. The Ontology is also mapped to procedures that are described at the Implementation Layer which allows information to be passed between the two levels based on the constraints prescribed in the logical processes.

As a process may be implemented differently in different system components, different versions of implementations may read, write, update or delete the same data sources concurrently following the explicit data management polices defined in the Logical Layer. The enacted processes may also communicate with each other through information that is under-pinned by the Domain Ontology. This mechanism enables a close collaboration between different process enactments and duplication of actions may be avoided and intelligent behaviours of the system may be generated.

The overall aim of the layered BPM framework is to provide a principled way for business process modelling that is flexible and therefore robust against changes in technology through time. It separates business requirements from technical issues when making decisions for developing workflow systems. This separation enables the workflow system to be more robust and agile in response to change of requirements in the dynamic environment that it operates within.

3 Requirements and Design of *FBPML*

To provide a business process modelling language that supports today's ever changing workflow environment and meets diversified requirements is not an easy task. A few design issues have been considered and acted upon, and are listed below.

- *Standard:* Modelling concepts that are described in the new BPM language should meet their specialised requirements but also need to be consistent with the current process modelling language standards. This not only keeps FBPML compliant with standard practices it also aids communication with other BPM languages and practitioners in the field. In essence, this means concepts that are included in standardised process modelling languages are main candidates to be included in FBPML. As a result, FBPML is an inherited, specialised and combined version of these standardised modelling languages. The main languages that have influenced the design of FBPML are IDEF3, PIF, PSL, RAD, CommonKADS CML and the Business Modelling method of IBM's BSDM.
- *Accessible:* The language should be easy to learn and use for both IT and business personnel. As one of the main business requirements for BPMLs is to enable business personnel to do BPM WITHOUT IT support.[12] To achieve this, FBPML covers *fundamental* process concepts that minimise complexity introduced by superfluous notations. It also introduces annotation notations that are informal and not directly understandable by machine. Such annotation is not formally a part of the model, but may provide useful explanation to the model, recording of design rationale or simply a reminder to assist the modelling process.
- *Collaborative:* An enterprise today is a virtual entity: it consists of a variety of enablers that are scattered across different geographical areas. Some enablers are human whereas others are autonomous agents or system components. Each enabler plays a role in its activities and is equipped with specialised functions, capabilities and authorities. Those enablers are characterised in their expertise and often behave in different ways that are best suited for their tasks and environment. However, to achieve organisational goals, they need to work collaboratively to accomplish their tasks. Traditionally, BPM methods do not include or explicitly represent the concept of such enablers, their responsibilities, authorities, how they collaborate with each other and what their relationships are between each other. The *roles* that enablers play, the relationships between them and information about them are captured in *FBPML* in the concept of *Role*.
- *Precise:* As most of the BPM methods are informal methods, they do not provide formal semantics for their notations. To avoid potential mis-use of the modelling language and mis-interpretation of built process models, there is a need for precise definition for notations so that a model may be interpreted correctly and consistently. IDEF3 provides a mature modelling method, graphical notations and sound conceptualisation about processes, but there is no formal semantic for its notation. PSL, on the other hand, does not have a visual presentation or method, but provides formal definitions of its concepts. This presents a natural opportunity to merge the two to gain benefits from both - this is the approach taken by FBPML.
- *Executable:* Semantics that are defined in the BPM language should include (or at least imply) operational definitions. This means the use of common process components, such as trigger, pre-conditions and postconditions, bear prescribed execution mechanisms. In addition, the types of executable activities also need to be identified and to be included as a part of the model. Process modelling methods are inherently rich in their semantics. The semantic of links between processes, for instance, are regarded as dependencies between processes, yet they also bear temporal constraints, and they may also act as triggers for the following processes. Junctions, such as AND, OR and XOR, may be interpreted differently depending on the use in the diagram, e.g. as a joint or split node. In addition, if both triggers and pre-conditions are defined in a process, they may bear distinct implications for

execution. Users of BPM need to understand such implications in order create a correct and appropriate model.

- *Formal:* Formality is important to connect a business process model to its execution phase. Ideally, there is a direct mapping from semantics of a business process model to application logic (as described in the logic layer and implementation layer in the previous section). This enables the separation between process and application logic, yet maintains declarative design of a workflow system. This implies modifications made at the logic layer automatically update processes at the implementation layer. If any inconsistency occurs, the system will give warning to the user. The formal approach has several advantages: automatic/intelligent analysis, verification, validation, and simulation facilities may be supported at the business layer[5][4]; once a business process model is satisfactory stable, it may automatically populate a large part of processes at the implementation layer.

4 A Declarative Executable FBPML - The Semantics

4.1 Activity, Decomposition and Specialisation

As mentioned in the previous section, FBPML should conform with standard practice. IDEF3, being a mature activity modelling method that largely meets our requirements, provides the foundation for *FBPML*. *IDEF3*[17] defines the concept of *decomposition* and *specialisation* of a process that *FBPML* also encompasses. Similar to *IDEF3*, the concept of *decomposition* in *FBPML* allows a process described at a higher level of abstraction to be decomposed into more detailed sub-processes that are more explicit for its implementation procedures. Each sub-process may also be decomposed into more detailed descriptions. The *specialisation* of a process indicates the alternative ways of carrying out a process.

Although there may be more than one alternative way of carrying out a task; unlike *decomposition* where all of the sub-processes must be carried out in order to accomplish the task, *specialisation* requires only one alternative sub-process to be carried out to accomplish the task. However, if one alternative activity does not finish the task due to some circumstances, another alternative activity may collaborate with the current one to accomplish the task. The detailed mechanism about how different alternative processes may work together in a coherent way in all eventualities requires a thorough examination of implementation methods. Since this is implementation dependent and outside the scope of this paper, it is not discussed here.

4.2 Notation

Figure 3 depicts the notation of FBPML as it is shown using KBST-EM (Knowledge Based Support Tool for Enterprise Models)[3]. There are three types of nodes: the *Main Node*, *Junction* and *Annotation*. Four types of *Main Nodes* are included: *Activity*, *Primitive Activity*, *Role* and *Time Point*. Two types of *Annotations* are included: the *Idea Note* and *Navigation Note*. Two types of links are provided: the *precedence-link* and *synchronisation-bar*. There are four types of *Junctions*: *and*, *or*, *start* and *finish*.

Main Nodes: As mentioned earlier, an *activity* node denotes the *type* of process that may be decomposed or specialised into sub-processes. In addition, the notion of *Primitive Activity* (from *PSL*) has been introduced to denote a *leaf node activity* that may not be further decomposed or specialised. Primitive activity is useful to FBPML, as it highlights the connecting point between the higher

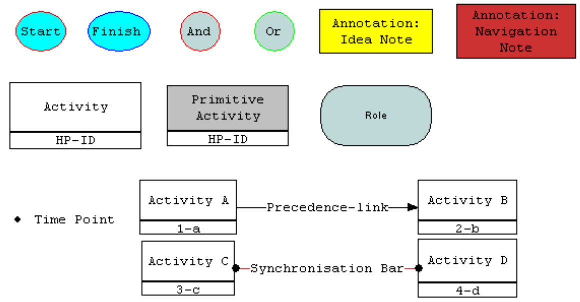


Figure 3. FBPML Notation

level process description and lower level implementation details that are described in the logical and implementation layers, respectively.

Although some process modelling methods distinguish terms between process, activity and task, as one is a higher level description of another, like IDEF3 and PSL *FBPML* does not make the distinction. Since a process may be further decomposed or specialised into sub-processes that may be again further decomposed or specialised, a process at one level is an activity to its "parent" process. As a result, these terms are used interchangeably in this document.

In FBPML, an activity is uniquely identified by its name (or ID)[1]. However, since FBPML (as well as *IDEF3*) permits the same activity to be repeated in different places *in a process model*, that normally exhibits different relationships between itself to other activities, the same activity may be enacted differently in a model in different places. Furthermore, since an activity may be a decomposition or a specialisation of its parent activity, this adds extra meanings depending on the type of sub-activity that it describes.

The *semantics of an activity to a model* is, therefore, defined together by *its location* in the model, *its usage* in the model and the *content* defined within itself, i.e. the *Trigger(s), Pre-condition(s)* and *Action(s)*. *Post-condition(s)* is often defined as a part of a process and recorded in our model as it gives explicit checking points on successful execution of a process. However, since it is derivable from pre-conditions and actions of a process, we do not include it in our formal representation. In FBPML, the location of an activity is recorded in the field *Hierarchical Position (HP)*. Therefore, the tuple

$$< HP, Activity\_name, Trigger, Pre\_condition, Action >$$

defines an activity (type) in a model using FBPML, where each HP is unique and there may be more than one trigger, pre-condition and action. To denote the relevance to and uniqueness in a model, an activity is formally represented as:

activity(Activity_name, Hierarchical_Position)

where Activity_name is the name of the activity and Hierarchical_Position its location in the model. If A is a *primitive activity* in the model, the above predicate name, activity, is changed to primitive_activity. Since this paper only discusses semantics of notations but not their semantics in a model, for simplicity, this section assumes all activities are uniquely used in our examples and therefore

[1] For pragmatic reasons, an activity ID is created for each activity to provide a short hand identity for an activity. Each activity name uniquely maps to an activity ID and vice versa. Logically, we do not represent it, since it does not add additional semantics.

uses Activity_name instead of the above predicates, activity/2, when referring to an activity.

The predicate attribute(Activity, Attribute_name, Attribute_value) holds the specification for an Activity type where Attribute_name stores the corresponding attribute name, such as trigger, precondition and action, and Attribute_value stores the attribute value that may be a structured term or template with variables using specific grammar. Variables that are included in the Attribute_value will be instantiated dynamically by (process or object) instances at run time.[2]

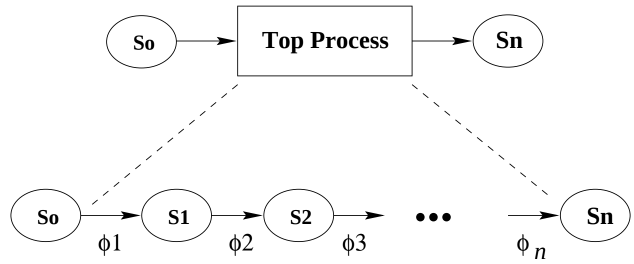
The concept of *Role* is adapted from RAD where a *Role* is described as involving a set of activities which carry out a set of responsibilities. Such activities are "generally carried out by an individual or group within the organisation". Roles are also types and "there can be a number of different instances of a role type active at any one time within an organisation"[19]. In *FBPML*, the definition of *Role* is functional and as described above, it defines the "role" that an enabler plays in the context of the described activities. Upon process enactment, a role may be fulfilled by an individual, a group of people or software components, or a combination of the above. Similar to *RAD*, although different graphical presentation and process concepts are used, *FBPML* highlights interactions between roles: each role may have its own internal as well as communication processes. The communication processes allow explicit definition of interaction methods and boundary of communication within processes of each role. Tasks and issues may be delegated, escalated or transfered between roles as a part of communication processes.

The notation of *time point* indicates a particular point in time during the enactment of a process model. The reference of time point may be decided by the implementation method of the model. A duration of a time interval is indicated by two time points. A length of time may not have association with any particular point of time.

Annotations: Two types of annotations are included: *Idea Note* records textual information that is relevant to, but outside the scope of, a process model, e.g. design rationale or a reminder for analysis for certain parts of a model; *Navigation Note* records the relationships between diagrams in a model. In general, annotation nodes do not contribute semantically to a process model, but they help the organisation and management of the modelling process.

Links: Two types of links are included: *Precedence-link* and *Synchronisation Bar*. *Precedence-link* is comparable to the more constrained Precedence Link, type II, in *IDEF3*. In FBPML, the specification that Activity A is preceded by Activity B is denoted by a *Precedence-link* from Activity A to B as shown in Figure 3. A Precedence-link places a temporal constraint on process execution that the execution of Activity B may NOT start before the execution of Activity A is finished when the two processes are on the same execution path. Figure 4 illustrates the concept of path and the execution of processes[1].[3]

In Figure 4, "Top Process" transforms from state So to Sn. It is also a parent process that may be decomposed or specialised into sub-processes. One way to propagate from state So to Sn then is to activate the appropriate sub-processes and execute them along the state path $\Pi =< So, S1, S2, ...Sn >$ by activating the process sequence Φ (where several process instances may execute synchronised or not to transfer from one state to another). We denote an execution of process instances along a state path Π in the predicate activation/2:



State Path: Π **= <So, S1, S2, S3,... Sn>**
Execution Path: $\Phi = \{\phi 1, \phi 2, \phi 3, ... \phi_n\}$

Figure 4. Execution Path for Processes

$activation(\Phi, \Pi)$.

Given the execution path, one can formally specify the temporal constraint between activity A and B in the formula below:

Axiom 1: Temporal Constraint
$$\forall Activity\_a, Activity\_b.$$
$$preceded\_by(Activity\_a, Activity\_b)$$
$$\Rightarrow$$
$$\left(\begin{array}{l} \forall A, \forall B, \exists P, \exists Act.instance\_of(A, Activity\_a) \wedge \\ instance\_of(B, Activity\_b) \wedge \\ activation(Act, P) \wedge \\ A \in Act \wedge B \in Act \\ \Rightarrow \\ end\_time(A, activity\_a) =< \\ begin\_time(B, activity\_b) \end{array}\right)$$

A *Precedence-Link* suggests natural process flow which is if Activity A is executed, Activity B should also be executed along the corresponding execution path unless other conditions interact with it. We use $\rhd$ to represent this nature of weaker inference that is pronounced as *should be* or *may be*. This definition gives a process model more flexibility and is slightly different from Precedence-Link Type II in IDEF3 where strong inference is prescribed. This rule is described formally below:

Axiom 2: Dependency Constraint
$$\forall Activity\_a, Activity\_b,$$
$$preceded\_by(Activity\_a, Activity\_b)$$
$$\Rightarrow$$
$$\left(\begin{array}{l} \forall A, P, Act.instance\_of(A, Activity\_a) \wedge \\ activation(Act, P) \wedge \\ A \in Act \\ \rhd \\ \exists B, instance\_of(B, Activity\_b) \wedge \\ B \in Act \end{array}\right)$$

A *Precedence-Link* also indicates that the completion of activity A invokes Activity B to be activated. We introduce a property **Temporal Qualification (TQ)** to denote that Activity B is temporally qualified to be executed. *Temporal Qualification,* however, does not guarantee the execution of an activity because it also depends on the content of trigger and pre-conditions of that activity. We use the predicate *tq(Instance, Process)* to indicate this property and *end/2* to indicate that the execution of a process instance is finished.

[2] A separate predicate is used to store process instance attributes.
[3] This Figure is adapted from [15].

Axiom 3: Property of Temporal Qualification

$$\forall Activity\_a, Activity\_b,$$
$$preceded\_by(Activity\_a, Activity\_b)$$
$$\Rightarrow \left(\begin{array}{l} \forall A.instance\_of(A, Activity\_a) \wedge \\ end(A, Activity\_a) \\ \Rightarrow \\ \exists B.instance\_of(B, Activity\_b) \wedge \\ tq(B, Activity\_b) \end{array} \right)$$

The property of *TQ* is important as it implies execution logic of a process model that separates the notation between the execution of process instances and those that are only temporally qualified to be executed. We introduce a separate property **Full Qualification (FQ)** to define that a process is *Fully Qualified*, if it is *Temporally Qualified* and that all of its triggers and pre-conditions are satisfied. A fully qualified process instance may be executed immediately. Due to space, we do not describe the formalism here. The properties of *TQ* and *FQ* provide exact semantics for the execution logic that determines the dynamic behaviours of a process model at run time.

The above precise definition of *FBPML* links signifies how it differs from most other business process modelling languages. Since most business process modelling languages focus on the specification ability of a process, the actual implementation steps of a process are left out and are open to interpretation for system developers, e.g. IDEF3, IDEF0, PSL, Business Process Model in BSDM. Since the implementation considerations have not been provided by the original model, it leaves a question of whether the implemented system obeys the intended design of the system and/or whether the implementation has been carried out consistently with respect to the model. Since such process execution rationale has not been recorded at the first place, such questions are difficult to evaluate.[4]

Besides providing precise execution logic and instructions to the implemented workflow system, the above precise semantics allows both static as well as dynamic (state) Verification, Validation and Critiquing (VVC) facilities on the business process model. The static VVC techniques include error and appropriateness checking and critiquing based on the examination and comparison of different parts of the static structure of a business process model without the actual instantiation of the model. The dynamic VVC involves test runs of interesting scenarios through the model in an attempt to understand system behaviours at run time. Similar techniques have been applied and implemented in *KBST-BM*[2] for IBM's business model in BSDM.

As an activity may be decomposed into several sub-processes, the activation of a top process may be accomplished by activation of its sub-processes. In this case, the execution of the top process is not finished unless all of the corresponding sub-processes are finished. Again, we do not describe the formalism here.

The second type of link is *Synchronisation Bar*. A Synchronisation Bar places a temporal constraint between two time points. For example, one may synchronise the starting or finishing of two processes by synchronising the "begin times" or "end times" of the two processes. The Synchronisation between two time points is therefore defined below:

$$\forall A \in time\_point, B \in time\_point.$$
$$synchronisation(A, B) \Leftrightarrow A = B.$$

Junctions: Junctions are special or simplified activities, in that they do not have triggers and pre-conditions, and their actions have predetermined decision logic for starting, ending or branching process execution. Four types of Junctions are included in FBPML: *start, finish, and,* and *or* junctions.

The "start" and "finish" junctions provide an explicit indication of the logical starting and finishing points of a process. They may also isolate a part of a process that can be treated locally as a sub-process. These two junctions provide a clear indication for the entry and leaving points for the reader and when executing a process. It provides a natural decomposition for testing a process and a convenient indication for breaking a long complicated process when developing workflow systems using a divide-and-conquer strategy.

An "and" or "or" junction is a *one-to-many relationship* that describes process execution flow and temporal constraint between the activities that are connecting to it. Figure 5 shows how an "and" or "or" junction may be used in a process model. As shown in the figure, there are two types of interpretations of an "and" or "or" junction: the *joint* or *split* type of junction, depending on the topology of the process model.

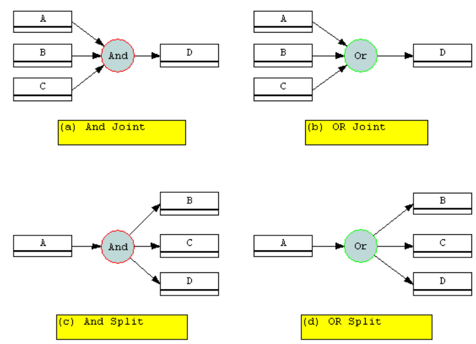


Figure 5. FBPML Joint and Split Junctions

An *and-* or *or-joint* indicates more than one preceding activity before the "and" or "or" junction, and only one activity following the junction. Figure 5(a) and (b) give example graphical representations of an and- and or-joint where each junction is attached to three in-coming arrows and only one out-going arrow. A joint type of junction is sometimes also referred to as a *fan-in* junction in some process modelling languages. Semantically, an and-joint indicates the process execution flow and the temporal constraint that *all* of the *preceding* activities must be finished before the following activity is temporally qualified and therefore be executed. An or-joint indicates *only one* of the *preceding* activities is required to be finished before the following activity becomes temporally qualified and executed.

An and- or or-split indicates that there is only one activity preceding the junction, but there is more than one activity following the junction. Figures 5(c) and (d) illustrate example and- and or-splits. A split junction is sometimes also referred to as a *fan-out* junction in some process modelling languages. Semantically, a split junction indicates process flow, temporal as well as dependency constraints. An and- or or-split indicates that *all* of the following activities become *temporally qualified* when the preceding activity is finished. Furthermore, an and-split also indicates that *all* of the *following* activities must be executed at some point of time after the preceding activity is finished.

On the other hand, an or-split indicates that *at least one* of the following activities of the "or" junction will be triggered and executed

[4] This is a recurrent problem that the authors have to deal with in one of their commercial business process modelling projects and their research projects.

when the preceding activity is finished. It is, however, unclear how many or which of the following activities will be triggered and executed, since it depends upon the corresponding dynamic system state and the trigger and pre-condition statements of the following activities. For both of the and- and or-split, all of the activities that are described after the junction may be executed in parallel or sequentially, when appropriate. The precedence-link and the junction do not specify the exact synchronisation between these activities. Such synchronisation is specified by *Synchronisation Bars*.

4.3 Combinational Use of Branching Junctions

Figure 6 demonstrates the four common combinational uses of “And” and “Or” junctions. The four basic cases of combinations are given in the Figure (a), (b), (c) and (d) accordingly and listed below: And-And, Or-Or, And-Or, Or-And.

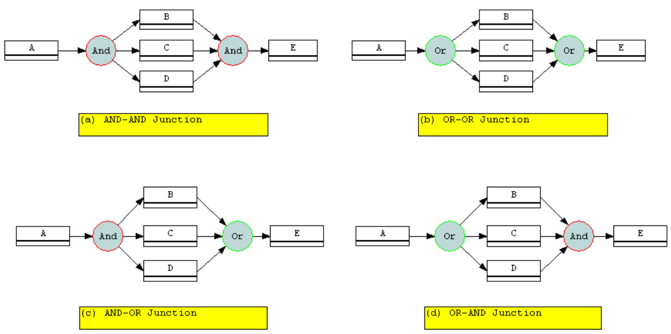


Figure 6. FBPML Junctions Coupled

According to the definitions given for “And” and “Or” junctions in the previous section, the and-and combination defines that activity B, C and D must execute at some point of time after but only after activity A is finished, and that activity E may not start execution before B, C and D have finished.

The or-or combination, on the other hand, gives a more loose constraint in that, similarly to and-and combination, activity B, C or D may only start execution after activity A is finished. However, it may not be the case that all of B, C and D are executed - it depends on the system dynamics and execution requirements of B, C and D. Nevertheless, since an or-split has been used here, at last one of B, C or D must be executed. When either activity B, C or D is finished, activity E may start its execution. The and-and and or-or combinations are demonstrated in Figures 6(a) and (b), respectively.

Similarly, in Figure 6(c), the and-or junction indicates that activities B, C and D may start their execution after activity A is finished, and activity E may start execution as soon as one of activities B, C or D is finished. What is different compared to Figure 6(b) is that activities B, C and D *must all* be executed at some point of time due to the and-split.

Figure 6(d) indicates that *at least one* of the activity B, C or D may be triggered and start execution after activity A is finished. Activity E may not start its execution unless all of the triggered activities, i.e. a combinations of B, C and D, are finished. Note that since an or-split has been used earlier in the process model, it may not be the case that all activities B, C and D are triggered. Nevertheless, all of the triggered activities must be finished before activity E may start its execution.

4.4 Discussion

As it has been described, an “And” or “Or” junction indicates a temporal constraint between the execution of connected processes. Furthermore, they also indicate the “execution” constraints that have been put in the process logic. For instance, an “and-split” indicates that all of the following activities must be executed when the preceding activity is finished. However, the model may not specify that all of the activities must be finished before the “next wave of activities” are started. One such example is given in Figure 6c, the case of and-or junction. Activities B, C and D may start execution in parallel but asynchronously and may finish their execution at different times. Activity E may start execution, as soon as one of them finishes execution. This means that activity E and activities following it may be executing along side the un-finished activity B, C or D. Furthermore, it is possible that all of the following activities after E are finished before activity B, C or D are finished. This may lead to an un-desirable result in the system.

The process model described in Figure 6c, however, is correct and appropriate when describing a situation where the start and execution of activity E is not temporally and semantically bound by activity B, C and D. However, when there is such a constraint at a later stage of the process that requires the finishing of the corresponding activity B, C or D, a limitation may be described in the triggers or pre-conditions of other following activities in the model.

One way to control and avoid “left-over” processes lingering indefinitely in the system is to define a process that is not finished until all of its (“left-over”) sub-processes are finished. Under this definition, the higher level process is not finished unless all of its sub-processes are finished. This is what has been defined in FBPML. Another way to control this is to provide a checking, alarming and repairing mechanism that will be triggered when processes are found lingering longer than a pre-determined period of time.

4.5 Demonstrating Dynamic Behaviours in Process Panels

As a part of the AKT project[6], for AKT-TIE[5], we have developed a small PC configuration business process model that accepts customer enquires and returns with possible pc-configuration specification. A snap shot of the business process model for the role “Edinburgh” is given in Figure 7 as it is shown in our support tool *KBST-EM*. This model has been successfully translated and displayed in a workflow stepping system, *I-X Process Panel*. Upon instantiation, instances of processes appear and are managed in *I-X system’s process panels*[24][16].

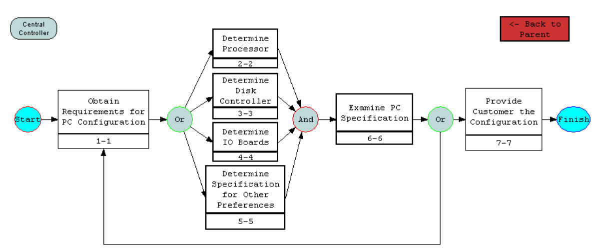


Figure 7. PC Configuration Business Process Model

[5] AKT-TIE is a part of the AKT project collaborating with Peter Gray and Kit Hui, Computer Science Department, Aberdeen University, UK.

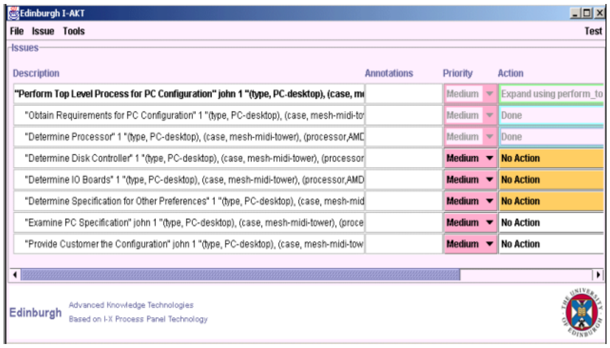


Figure 8. View of I-X System Process Panel

Figure 8 demonstrates how the instantiation of each process presents an entry in the I-X process panel. Each entry consists of two components: the name of the process and variables the process takes. The parent process of processes given in Figure 7, “Perform Top Level Process for PC Configuration”, is shown at the top row and in bold face which is decomposed into sub-processes as those described in Figure 7.

In I-X, for each process instance, several actions may be performed upon them and the execution status of each instance is reflected by different colours. In I-X, all process instances may be executed (done), cancelled (Not Applicable), waiting to be processed (No Action (yet)), or decomposed into sub-processes (Expansion). Communication processes in our model may also dispatch tasks to other appropriate “roles” as defined in their processes. Branching of processes is controlled by the availability of actions that may be performed on the instances. For instance, in Figure 7 all processes on the second column of the model that are after the or-split may be executed in parallel, but this operation is only available after the “Obtain Requirements for PC configuration” process completed its execution.

It has become apparent that it is not an easy task to provide a declarative BPML that provides direct support for building and executing workflow systems and that more issues are to be investigated and resolved. Typical action types should be provided by the languages so that any models built using the language benefit directly from it, while at the same time one needs to allow flexibility and ease for addition or modification on existing action types. To safeguard against inconsistencies at the modelling language level is to provide some form of (automatic) inconsistency checking on static models and dynamic environments. Upon executing a process model, it is also vital that static processes are provided but the workflow system must be able to allow the users to dynamically modify or add new processes. Again, this will have to be done within a predetermined safety level.

5 Conclusion

Enterprise Models need to bridge the gap to software system development and execution, but additional mechanisms are needed so that information that is held within them may be transferred and mapped onto software execution. To bridge this gap, however, is not a minor task. Diverse and often conflicting requirements are need to be addressed. In addition, formality needs to be introduced to the informal or semi-formal enterprise modelling paradigm to provide precision and enable automatic support. When domain knowledge is used as a part of software system development and execution, it is also needed to ensure that it has been checked for consistency and appropriateness during the phase of enterprise modelling. This paper proposed a declarative modelling approach in an attempt to bridge the gap between business process modelling methods to (workflow) software systems.

Based on this approach, an initially static, high level business process specification may be represented formally and automatically. Based on the formalism, automatic verification, validation and critiquing may therefore be provided as a part of normal modelling activities. Furthermore, the modelling notation bears exact execution instructions that may be mapped to software modules that are components of a workflow system. This gives the prospect of rapid prototyping and testing of a workflow system that is based on the model. This benefit will not be possible without providing execution semantics in a model.

It will be advantageous that more similar work as reported in this paper is carried out for all Enterprise Models to narrow the gap which currently exists at various places between EM methods and software system development. When this is done, the set of Enterprise Models together may provide a holistic and clearer view as well as more direct instructions, particularly from the business, organisational, knowledge, information and process points of view, to assist the process of software system development for the organisation.

Acknowledgement

This work is carried out as a part of the Advanced Knowledge Technologies (AKT) (IRC) project[6], which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. The EPSRC and the Universities comprising the AKT IRC are authorised to reproduce and distribute reprints for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of the EPSRC or any other member of the AKT IRC.

REFERENCES

[1] A. J. Bonner, ‘Workflow, transactions, and datalog’, *Proceedings of the Eighteen ACM Symposium on the Principles of Database System (PODS), Philadelphia, PA.*, 294–305, (May 1999).

[2] Yun-Heh Chen-Burger, *Formal Support for an Informal Business Modelling Method*, Phd thesis, Artificial Intelligence, The University of Edinburgh, 2001.

[3] Yun-Heh Chen-Burger, ‘Knowledge sharing and inconsistency checking on multiple enterprise models’, *International Joint Conference on Artificial Intelligence, Knowledge Management and Organizational Memories Workshop, IJCAI 2001, Seattle, Washington, USA.*, (August 2001). Also available as Informatics Division Technical Report, University of Edinburgh. In print as a chapter of book, publisher: Kluwer.

[4] Yun-Heh Chen-Burger, Dave Robertson, and Jussi Stader, ‘A case-based reasoning framework for enterprise model building, sharing and reusing’, *Proceedings of ECAI Workshop: Knowledge Management and Organizational Memories, Berlin.*, (August 2000).

[5] Yun-Heh Chen-Burger, David Robertson, and Jussi Stader, ‘Formal support for an informal business modelling method’, *The International Journal of Software Engineering and Knowledge Engineering*, (February 2000).

[6] AKT Consortium. http://www.aktors.org, October 2000. Interdisciplinary Research Collaborations (IRC), Advanced Knowledge Tech-

nologies (AKT) Project. Partners: University of Southampton, Aberdeen, Edinburgh, Sheffield and Open University, UK.

[7] J.E. Dobson, A. J. C. Blyth, J. Chudge, and M. R. Strens, 'The ordit approach to organisational requirements', *Requirements Engineering: Social and Technical Issues*, (1994). London, ed. Jirotka and J.A.Goguen, Academic Press.

[8] *Workflow Handbook 2001*, ed., Layna Fischer, Future Strategies Inc., 2000.

[9] M. S. Fox and M. Gruninger, 'Enterprise modelling', *AI Magazine, AAAI press.*, 109–121, (Fall, 1998).

[10] Ulrich Frank, 'Multi-perspective enterprise models as a conceptual foundation for knowledge management', *Proceedings of Hawaii International Conference on System Sciences, Honolulu.*, (2000).

[11] Norbert. E. Fuchs and David Robertson, 'Declarative specifications', *The Knowledge Engineering Review*, **11**(4), 317–331, (1996).

[12] Delphi Group, 'Bpm 2002: Market milestone report', *Web site: www.delphigroup.com/ coverage/ bpm_webservices.htm*, (February 2002).

[13] IBM, UK, *Business System Development Method: Business Mapping Part1: Entities*, 2nd edn., May 1992.

[14] Stefan Junginger, Harald Kuhn, Mark Heidenfeld, and Dimitris Karagiannis, 'Building complex workflow applications: How to overcome the limitations of the waterfall model', *Workflow Handbook 2001*, 191–206, (2000).

[15] Michael Kifer, 'Introduction to transaction logic', *A tutorial presented at IPLS'97 (International Logic Programming Symposium), Port Jefferson, Long Island, N.Y.*, (October 1997). www.cs.sunnysb.edu/ kifer/dood/tr-tutorial.html.

[16] J. Levine, A. Tate, and J. Dalton, 'O-P^3: Supporting the planning process using open planning process panels', *IEEE Intelligent Systems*, **15**(6), (November 2000).

[17] Richard Mayer, Christopher Menzel, Michael Painter, Paula Witte, Thomas Blinn, and Benjamin Perakath, *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*, Knowledge Based Systems Inc. (KBSI), September 1995. http://www.idef.com/overviews/idef3.htm.

[18] National Institute of Standards and Technology, *Integration Definition for Function Modelling (IDEF0)*, December 1993.

[19] Martyn A. Ould, *Business Processes: Modelling and Analysis for Reengineering and Improvement*, John Wiley and Sons, 1995.

[20] D. Robertson and J. Augusti, *Software Blueprints: Lightweight Uses of Logic in Conceptual Modelling*, Addison Wesley, May 1999. in press.

[21] Craig Schlenoff, Amy Knutilla, and Steven Ray, 'Proceedings of the process specification language (psl) roundtable', *NISTIR 6081, National Institute of Standards and Technology, Gaithersburg, MD*, (1997). http://www.nist.gov/psl/.

[22] Jussi Stader and Ann Macintosh, 'Capability modelling and knowledge management', *Proceedings of Expert Systems 99, The 19th International Conference of the BCS Specialist Group on Expert Systems*, (1999). Springer-Verlay.

[23] A. Tate, 'Towards a plan ontology', *AI*IA Notizie (Quarterly Publication of The Associazione Italiana per l'Intelligenza Artificiale), Special Issue on "Aspects of Planning Research"*, **9**(1), 19–26, (March 1996).

[24] Autin Tate, 'I-X: Technology for intelligent systems', *www.i-x.info, AIAI, The University of Edinburgh*, (2002).

[25] Mike Uschold, Martin King, Stuart Moralee, and Yannis Zorgios, 'Enterprise ontology', *The Knowledge Engineering Review: Special Issue on Putting Ontologies to Use*, **13**, (1998). Also available as technical report from AIAI, The University of Edinburgh (AIAI-TR-195).

[26] Annika Waern, Kristina Hook, Rune Gustavsson, and Peter Holm, 'The common-kads communication model', Kads-ii/m3/sics/tr/006/v2.0, Swedish Institute of Computer Science, Stockholm, Sweden, (December 1993).

A Cooperative Approach to Corporate Memory Modeling

Jesualdo Tomás Fernández Breis[1], **Rodrigo Martínez-Béjar**[1],
Laura María Campoy-Gómez[1], **Fernando Martín-Rubio**[1], and **Juan José García-Martínez**[2]

Abstract. Nowadays, the importance of Knowledge Management is growing in organizational contexts. Corporate Memory is an appropriate tool to represent organizational knowledge. This work presents an ontology-based approach to Corporate Memory modeling. In it, the members of an organization act as 'knowledge builders' and they construct the Corporate Memory co-operatively. Furthermore, the employees who take part of the Corporate Memory construction process are allowed to use their own terminology, even for requesting information about the Corporate Memory until a specific instant.

Keywords: Knowledge Management, Ontology, Corporate Memory, Mereology, Reusable Component.

1 INTRODUCTION

Nowadays, Knowledge Management (KM) is one of the key factors in organizations since the current trend is to evolve from employees to 'knowledge workers' ([6]; [21]). The fact is that organizations are realizing that knowledge increases the value of their products and services in addition to providing a competitive advantage. According to [29], the objectives of KM in an organization are to promote knowledge growth, knowledge communication and knowledge preservation in the organization. There are various types of significant knowledge for an organization. Thus, we should mention the identification of critical knowledge functions and the knowledge of who knows what in the organization as the most important factors. This knowledge must be kept in some way in the organization and that is why the concept of Corporate Memory (CM) arises.

The know-how knowledge is usually distributed inside an organization, so in order to facilitate its access and reuse it must be integrated coherently, that is, expressed as a CM. This has been considered as a key element for performing Knowledge Management because it facilitates knowledge conservation, distribution, and reuse. In recent literature we can find many definitions for CM. The authors in [31] defines a CM as an "explicit, disembodied, persistent representation of knowledge and information in an organization" while [23] does it as "the collective data and knowledge resources of a company, including project experiences, problem solving expertise, etc".

[1]Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia, CP 30071, Murcia, Spain.

Emails:{jfernand,rodrigo, fmartin}@dif.um.es, campoyl@um.es

[2]Departamento de Matemática Aplicada, Universidad de Murcia, CP 30071, Murcia, Spain, email:jjgarcia@um.es

In [2] a CM is understood as "a container that integrates contextual information, documents and unstructured information, facilitating its use, sharing and reuse". Its main function would be to improve the organizations' competitiveness through the way their own knowledge is managed. Some authors consider the CM as a link mechanism between past and future knowledge in relation to the processes and activities that take place inside organizations. In particular [30] defines CM as "the means through which all the operative knowledge accumulated in the past is put in the present to be used in the activities performed in the organization".

In [8], the authors came up to the concept of Corporate Memory through the concept of activity in an enterprise. For these authors, a CM is comprised of a set of activities and an activity is defined as in [18]: "what people do, hour after hour, day after day: finally, employees achieve all these works because they know they can do them, they think they have to do them, all of this doings involving specific know-how as simple as they could be".

Regarding the activitities involved in KM as a process, according to [11] the following ones can be enumerated: i) identification and mapping of intellectual goods belonging to the organization, ii) generation of knew knowledge that will permit gaining a competitive advantage, iii) compilation of amounts of organizational information in an accessible way, and, iv) sharing best of practice and technology, including groupware techniques and the intranets.

In [26], the CM management is described as comprised of six processes: detection of needs, building, distribution, use, evaluation and evolution of the CM. Our work addresses four of these steps: knowledge construction, knowledge distribution, use of the knowledge and maintenance of the knowledge. Our approach to the problem has been carried out through a distributed perspective, that is, we have defined a system to manage Distributed Corporate Memories which facilitates knowledge sharing and collaboration between groups of people which can be at different geographic locations.

The technology used to represent the knowledge in our work has been the ontology, element that has been considered to be necessary to perform an appropriate Knowledge Management ([4];[25]). An ontology is commonly viewed as a specification of a domain knowledge conceptualization [32]. We can find domain ontologies (for example, a virus ontology in medicine) and enterprise ontologies (description of an enterprise model). Both of them can and must be included in a corporate memory. In our approach, each group of people generates an ontology concerning the previously mentioned significant (for the organization) knowledge of the group. This ontology represents a part of the organization, which must be shared with the rest of groups that belong to the same organization or to other collaborating organizations. To allow this knowledge sharing, we must proceed to integrate the knowledge from each ontology (one for each group). The cooperative building of knowledge pieces is an emergent topic and there are also different projects working on it such as KA^2 [3], Chimaera [20] or PROMPT [16].

The structure of this article is the following. Section 2 introduces some ground concepts of our approach, as well as how the system faces the steps of Corporate Memory and Knowledge Management. In Section 3, we explain the ontological model followed to represent the knowledge of each group of the organization. Section 4 describes the main characteristics of the system implemented. Section 5 presents a practical example of an application domain through which the system has been validated. Finally, we make some final conclusions in Section 6.

2 KNOWLEDGE MANAGEMENT AND CORPORATE MEMORIES

The main processes in KM can be described as in [11]: "identifying and mapping intellectual assets within the organization, generating new knowledge for competitive advantage within the organization, making vast amounts of corporate information accessible, sharing of best practices, and technology that enables all the above, including Groupware and intranets". Therefore, it cannot be seen as a product but as a process which has to be implemented over a period of time. As it is pointed out in [4], this process has "as much to do with human relationships as it does with business practice and information technology".

Distributed Knowledge Management Systems (DKMSs) are increasing their significance rapidly due to the growing importance of knowledge distribution. An example of a DKMS is a Corporate Memory (CM). A CM integrates contextual information, documents and unstructured information, facilitating its access, sharing and reuse. Its main function is to enhance the organization's competitiveness by the way it manages its knowledge [1].

In this work, we assume that an organization is divided into different groups. Each group is comprised of people and a group can be characterized by its number of members. Groups can have one or more members, so that those can be described as non empty sets of persons. From a functional point of view, a group can be a department of the organization or a group of people in charge of some specific tasks or responsibilities in the organization because not all the organizations or enterprises are organized in the same manner. Therefore, our notion of group was conceived to be flexible enough to be applied to a variety of types of organization structure.

The concept of 'group' is not new in the context of Corporate Memories. It has already appeared in literature, for instance in [14], "a CM is a repository of knowledge and know-how of a set of individuals working in a particular firm". Our concept of CM is not restricted to a unique organization but it is also applicable to a co-operative enterprise. For us, a co-operative enterprise can be seen as a collection of groups of people belonging to one or more organizations, so that those groups can and must work co-operatively.

Figure 1 shows our concept of co-operative organization as well as the division of the organization in the terms that this work has been focused on. In this particular case, there are two organizations divided into groups. Each organization is comprised of two groups and each one has a manager. Organization 1 and Organization 2 have made an agreement for collaborating in doing some project. Therefore, the co-operative organization is comprised of four working groups and the corporate memory for this organization must cover the knowledge generated by the four groups. The manager has only local significance and there must be a decision about who is going to be the manager of the organization. The graphic on the right side of Figure 1 represents the internal structure of a group in the organization. A group is defined as a set of employees and an administrator who manages the group.

2.1 Corporate Memory Management

As we pointed out before in this paper, the management of a Corporate Memory is comprised of six main steps: detection of needs in the organization, construction of the CM, diffusion of the CM, use of the CM, and evaluation and evolution of the CM [10]. The first step is out of the scope of this work since it requires an exhaustive analysis of the organization in order to establish its needs and our approach tries to be generic and organization independent. The rest of steps have been dealt with in our work. In the following sub-sections, we present how our system performs the mentioned steps.

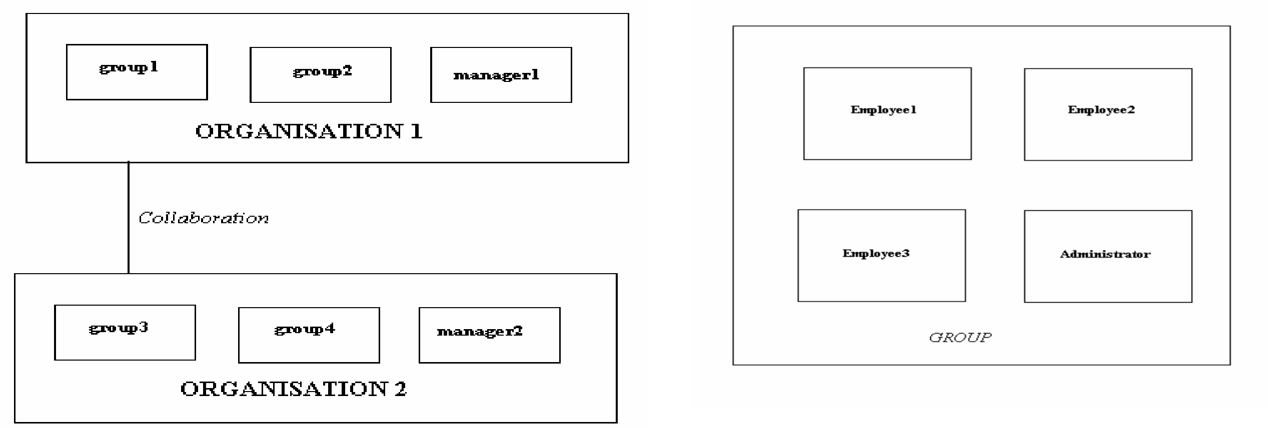


Figure 1. A co-operative organization

2.1.1 Construction

The Corporate Memory is built from the knowledge that exists in the organization. According to [6], knowledge has a double nature: explicit and tacit. Explicit knowledge is the knowledge that can be explained verbally or written down easily. On the other hand, tacit knowledge is the knowledge that cannot be made explicit due to different reasons, such as the impossibility of making certain knowledge available for others, (i.e., the incapacity of externalizing it in order to make it explicit).

We can find four different patterns for the creation of knowledge in an organization [24]:

- Socialization: Sharing tacit knowledge between individuals. The knowledge remains tacit without being transformed into explicit. This kind of pattern is not very interesting for the organization because of its tacit nature. (Tacit → Tacit)
- Articulation: Someone transforms tacit knowledge into explicit knowledge. (Tacit → Explicit)
- Synthesis: Combination of explicit knowledge to create new explicit knowledge. (Explicit → Explicit)
- Internalization: Process of transforming explicit knowledge into tacit knowledge. (Explicit → Tacit).

In our approach, we are only interested in knowledge expressed in an explicit manner, because it is the unique type of knowledge that can be directly (i.e., without processing) shareable, accessible for and reusable by people within the same group or organization. The knowledge is created by the employees of the organization, who are members of one or more groups of the (co-operative) organization. All the knowledge is made explicit by some specialized applications integrated in our system, so becoming shareable and reusable in an easier way. Our choice for internally representing knowledge has been the ontology as we stated above and the ontological model followed to develop this system is described further in this paper. In summary, the knowledge is created by employees and put into the CM by the system through ontologies because of the good properties of ontologies for facilitating CM Management. These allow for knowledge sharing and reuse, in addition to the ontology characteristic of permitting a formal representation of knowledge. This ontology feature is another key factor when deciding which representation technology is the most appropriate for knowledge modeling.

To end with the creation of knowledge, we should mention the facilities to express knowledge offered by the system. The basic knowledge element is the concept, which can be a logical or physical entity in the organization. Examples of organization-relevant concepts are department, employee, process, etc. These concepts have attributes (i.e., properties) that make them different from other concepts, that is, a concept is partly characterized by its attributes, although it is also characterized by its relationships with other concepts of the corporate memory. In this work, two types of attributes are considered:

- Specific attributes: These are the attributes a concept has by its nature.
- Inherited attributes: These attributes are derived from relationships with other concepts.

Concerning the relationships a concept may have, we contemplate three types of inter-concept relationships:

- CLASS-OF: It means that a concept 'is a class of' another concept. For instance, an employee 'is a class of' person. This kind of relationship is useful to establish the hierarchies at different levels in the organization and it implies attribute inheritance. A concept is a classification of another concept attending to one or more attributes of the parent concept. This non-empty set of attributes of the parent concept by which the classification is made is called the 'specialization' that every CLASS-OF relationship induces.
- PART-OF: It means that a concept 'is a part of' another concept'. For instance, an employee 'is a part of' a department. Partonomies are useful to express structural divisions in the organization or in elements of the organization (departments, processes, etc).
- AFTER: It means that a concept 'occurs after' another concept. For instance, the process of evolution of knowledge 'occurs after' the process of evaluation of the knowledge. This kind of relationship is important in an organization to establish temporal links between processes. For example, if we are modeling the resolution process of a failure, there can be different tasks to perform in order to fix it. This process will involve a task execution order that may be established by using this kind of relationship.

These three types of relations are included in the set of most common relations in real domains [17]. In particular, we consider the temporal relation useful in organizational contexts in order to deal with workflow representation and management (i.e., the temporal dimension underlying organizational processes can be specified through this type of relation).

2.1.2 Distribution

This aspect concerns the distribution of knowledge to the staff of the organization. In particular, the purpose is to know who is allowed to know what in the organization. If the distribution is made automatically this will occur as soon as new knowledge is available or after a request for knowledge actualization is made. However, the distribution process has two groups of elements that take part in it. The first group is comprised of the groups (or employees) who have new knowledge to introduce in the corporate memory, that is, people who can communicate some organization-relevant knowledge in some way to the rest of the organization. The system must capture this knowledge first in order to make it available for the rest of the community (the co-operative organization in this case). The second group is formed by the rest of the mentioned community, namely, people who must be interested in having access to the new knowledge available in the organization. Therefore, knowledge distribution can be regarded from two perspectives: knowledge collection versus new knowledge access.

Knowledge collection is a more critical factor for us, so it must be performed on a 'as soon as possible' basis, that is, when the system detects or assumes the existence of new knowledge, it must be retrieved. Thus, when employees are generating knowledge for the organization and other employees want to check for the existing knowledge, the system must retrieve the new knowledge in order to provide the best possible knowledge to the employees who request for that knowledge. The discussion about knowledge collection can be moved to a different domain, namely, knowledge distribution, which is concerning with how and when employees have access to the knowledge. An employee will be able to receive the new knowledge included in the corporate memory of the organization by requesting for it. Therefore, this process can be seen as a passive knowledge distribution. The knowledge created by the employees of the organization is stored in a knowledge server and the system provides a web-based access to the corporate memory via Internet/Intranet.

2.1.3 Use

A corporate memory management system must provide a simple and comfortable use for the employees of the organization. In other words, the exploitation process of the system must be conceived to be friendly with the system users (i.e., the employees of a (co-operative) organization). This implies to provide a well-documented system and friendly, intuitive user interfaces without forgetting that we are providing a web-based access to the knowledge. Another aim of our work was to display the information graphically. The exploitation of our system is briefly explained in Section 4, and the system's modus operandi is illustrated through a practical example (Section 5).

According to those requirements, the system has been designed to have flexible knowledge visualization, allowing the users to see what they want at each instant. To be more precise, the following visualization options are facilitated by the system:

- Complete corporate memory: This option shows the hierarchy defined by the corporate memory at a specific instant.
- Concept exploration: This option allows the user to visualize a specific concept, in terms of attributes and relationships with other concepts belonging to the corporate memory.
- Expanding taxonomic hierarchies: This option visualizes the existing taxonomies with respect to a specific set of attributes of a concept.

2.1.4 Maintenance

We can bring the processes of evaluation and evolution of the corporate memory together into the process of maintenance of the CM, although we can discuss about them independently. The evaluation of the CM means to estimate the usefulness of the CM for the organization from different points of view. The objective of this process is to assess the improvements originated by the introduction of the CM in the organization. The evaluation of the CM is out of the scope of this work because this process is organization-dependent. However, we think that the exchange of know-how within the organization will be always a benefit for it.

Concerning the evolution of the corporate memory, [11] stated that it depends on the results of the evaluation process. This is obvious because if the organization estimates that the corporate memory is useless for its purpose, there will be no need for maintaining the CM working. Maintaining a CM implies to add new knowledge when it is generated, to remove obsolete knowledge from the CM and to solve coherence and consistency problems which are intrinsic problems of co-operative work. The removal of obsolete knowledge can be made by the system manager, who can and must decide when some knowledge has become obsolete. Another possibility is that the obsolete knowledge is replaced by new knowledge belonging to the same user or group.

The addition of new knowledge to the system has been explained in the sub-section about knowledge creation, but we do not have explained what happens when the new knowledge is inconsistent with other existing knowledge in the corporate memory. The system we present here has a user-oriented philosophy for managing the knowledge a specific user is going to receive. That is, our knowledge integration approach makes it available to the user the integration of the knowledge kept in the system that is consistent with his/her own knowledge.

3 THE ONTOLOGICAL BASIS OF THE SYSTEM

In this section, the technology used in this work to make it possible the Corporate Memory Management is described. An ontology is seen as a specification of a domain knowledge conceptualization. Ontologies are represented here by means of *multiple hierarchical restricted domains* (MHRD) in a similar sense of that employed by other authors (see, for instance, [13]). The notion of *Partial, Hierarchical, Multiple and Restricted Domain* (PHMRD) [19] has been utilized for this work. A PHMRD can be specified as a set of concepts which are defined through a set of attributes. In PMHRD's, we contemplate three types of permitted relationships among whatever two concepts: taxonomic (allowing for multiple inheritance), mereological and temporal ones. Taxonomic relationships are assumed to hold all the irreflexive, the antisymmetric and the transitive properties, while mereological relationships are assumed to hold all of them except for the transitive one [5].

Regarding temporal concept relationships, these hold the same properties as taxonomic relationships. In order to implement this type of relationships, the FTCN model, as employed in [7], has been used. This model has been introduced to formalize the computational representation of general situations in which an arbitrary number of events are specified. A FTCN is a couple <X,L>, where $X= \{X_0, X_1, ..., X_n\}$ is a finite set of variables and $L= \{L_{ij} \mid i, j \leq n\}$ represents a finite set of binary fuzzy constraints. The variable X_0 represents a precise origin, in our case, when the time is supposed to start (i.e. when the first process of the temporal chain starts). Therefore, each constraint L_{0i} defines the absolute value of X_i. By translating this into the organization domain, if X_i stands for the occurrence of a specific process, L_{0i} will define the fuzzy time at which the process starts.

In this work, we have made use of possibility distributions for the FCTN model. In particular, the trapezoidal one has been employed, because of its good properties for our goal. We can characterize a trapezoidal distribution by four parameters: $\pi_j = (\alpha, \beta, \gamma, \delta)$:

- Base of the distribution: Set of values $t \in \tau$ such that $\pi_j(t) > 0$. It gives all the possible values.
- Kernel of the distribution: Set of values $t \in \tau$ *such that* $\pi_j(t) = 1$. It gives the completely possible values.

The left hand side of Figure 2 shows a generic trapezoidal distribution, while the right part of it shows the fuzzy number associated to an event whose occurrence time is "approximately at 8:00". Arithmetic operations on this distribution are reduced to apply them to the base and kernel, as follows:

$$1.Addition: (\alpha_1, \beta_1, \gamma_1, \delta_1) \oplus (\alpha_2, \beta_2, \gamma_2, \delta_2) = (\alpha_1 + \alpha_2, \beta_1 + \beta_2, \gamma_1 + \gamma_2, \delta_1 + \delta_2)$$

$$2.Intersection(\alpha_1, \beta_1, \gamma_1, \delta_1) \cap (\alpha_2, \beta_2, \gamma_2, \delta_2) = (max\{\alpha_1, \alpha_2\}, max\{\beta_1, \beta_2\}, min\{\gamma_1, \gamma_2\}, min\{\delta_1, \delta_2\})$$

Once we have introduced the complete network, the next task it to minimize it in order to find the minimal network that meets the original constraints. This will help us to calculate the estimated occurrence time of each process, which will be its absolute value from X_0, L_{0i}, as we stated previously. The algorithm that we have used detects inconsistency in the network and produces a minimal network as well. The body of such an algorithm is the following:

```
begin
  for k := 0 to n do
    for  i := 0 to n do
      for j:= 0 to n do
        Lij:=Lij∩(Lik⊕Lkj);
        if Lij = π∅ then exit "inconsistency"
end
```

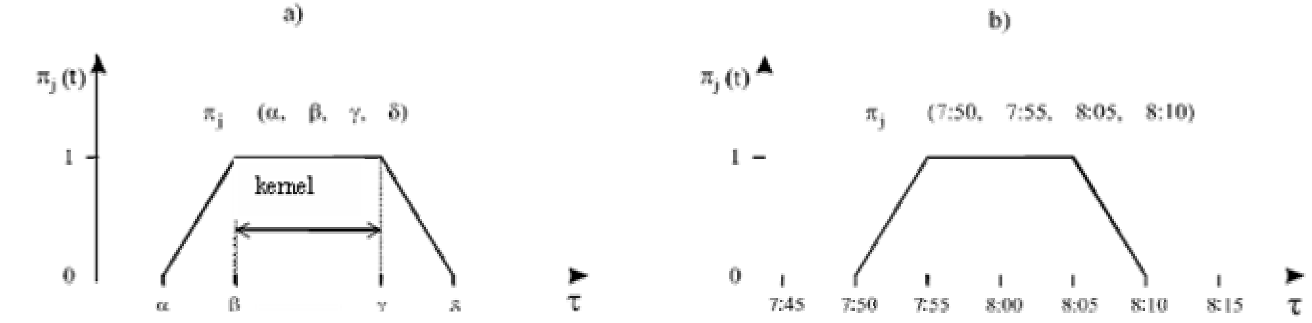


Figure 2. An example of trapezoidal distribution

We assume that the system is supplied with ontologies without inconsistencies in order to avoid the evaluation of them once they have been built. For it, and given that each ontology can be built in a particular way, users in charge of dealing directly with the (ontological) internal knowledge representation in our approach (i.e., employees of the KM department) must introduce their own ontologies by using a specific format for the ontology file. In this work, we refer to users as people belonging to the organization that are using the KM system presented here. The specification of this pseudo-language can be resumed as follows. It is comprised of the concepts which are part of the ontology. Each concept is defined through its attributes, its name and its parent concepts, either mereological, taxonomic or temporal ones. The successfully parsing of the ontologies defined according to this model is granted to be consistent.

3.1 The Integration Process

The author in [22] states that the reuse of ontologies has important advantages in Knowledge-based Systems research. We agree with that statement because it is easier to generate knowledge from different source ontologies (belonging to the groups) than generating it from scratch (i.e., starting from having no information at all). As we mentioned previously in this article, the aim of this work was the design and implementation of a tool for building distributed corporate memories from the knowledge supplied by a set of groups of people. Thus, the starting point for this process is a set of "individual" ontologies that have been built by employees or by groups. Thus, through the integration process the different existing ontologies are transformed into a global ontology which unifies the knowledge of each (different) viewpoint. Some considerations about such "individual" ontologies should be made. Such ontologies possess a private nature, that is, no one except for the owner has access to them, as well as the capability for modifying the knowledge included in those ontologies. Therefore, employees/groups have not direct access to other employees'/groups' knowledge, although they can benefit from the knowledge possessed by other employees/groups by means of the integration process. A user can only see his/her private ontology and the global one. Moreover, integration is performed transparently to knowledge suppliers, because they do not know when integration processes are requested and performed.

Due to the cooperative nature of the integration process, different problems must be solved, namely, redundancy, synonymy, and inconsistency:

a) *Redundant Information*. Two different ontologies might attempt to describe the same part of the domain knowledge. Given this eventuality, it would be desirable that the system was capable of managing this possible situation so that redundancies could be avoided.
b) *Use of synonym terms for a concept*. Apart from dealing with redundant information, different ontologies can employ different terminologies for the same concept. In other words, there can be a correspondence between different terms employed for a given concept [28]. During the ontology construction process, the information concerning the use of synonym terms for a concept must be stored and managed, since a particular terminology should not be imposed to any expert during the Knowledge Acquisition process. However, an ontology would strive towards 'consensual knowledge', that is, a fixed terminology. Synonyms are possible but, ideally, everybody should agree on the terminology. *Inconsistent knowledge*. This aspect has a double nature.
c) An ontology can be internally or externally inconsistent. We say that an ontology is internally inconsistent when there are some parts of it that are inconsistent with other parts of itself. For instance, an ontology is internally inconsistent if any property concerning relationships between concepts is not satisfied (i.e., if a concept 'A' is a taxonomic parent of another concept 'B' and 'B' is a taxonomic parent of 'A' then the ontology is internally inconsistent. An ontology can be externally inconsistent with respect to another ontology, that is, both descriptions of the same domain are incompatible. In particular, inconsistencies between (a part of) the knowledge corresponding to a group's ontology at a given instant (we refer to this ontology from now on as $O_i(t)$) and the knowledge of the ontology obtained by the integration process until that instant (we refer to this ontology from now on as $O_{int}(t)$) could appear. In this case, the knowledge from $O_i(t)$ would be assumed as the valid one, because we have considered the fact that $O_{int}(t)$ might have been checked by the owner group during the construction of their own ontology.

In order to achieve this goal, the system must be able to solve some possible consistency conflicts between the candidate ontologies to be integrated until a specific instant. In particular, each time that a group adds or modifies knowledge to its private ontology, such knowledge will have to be incorporated into $O_{int}(t)$. It is also remarkable that more than one group might decide to send its knowledge contribution to $O_{int}(t)$ at the same time. This made it necessary that the system was able to distinguish amongst pieces of knowledge belonging to different groups. In this sense, a group-oriented integration principle has been followed, which basically states that 'the knowledge in $O_{int}(t)$ at a specific instant will have to be consistent with that included in every private group ontology ($O_i(t)$) for every previous instant'.

In order to integrate the knowledge specified in groups' ontologies (i.e., the ontologies that belong to groups who are members of the same co-operative organization), the following algorithms have been followed [15].

3.1.1 Ontological_Integration

Let $O_i(t)$ be the i-th ontology that is intended to be incorporated into $O_{int}(t)$; n = number of ontologies to integrate. Let candidates(t) be the set of ontologies to integrate.

For i=1 to n
  If (there is any ontology $O_j(t)$ belonging to candidates(t) such that $O_i(t)$ and $O_j(t)$ belong to the same user) then (remove from candidates(t) the oldest ontology)
  End-if
End-for
subset= *Select_Ontologies*(candidates(t))
i=1
While i ≤ Card(subset) do
 Ontological_Inclusion($O_i(t)$,$O_{int}(t)$) (this algorithm is defined below).
End-while
Ontological_Transformation(*$O_{int}(t)$*)
End

3.1.2 Ontological_Inclusion

Let $O_j(t)$ be the j-th ontology that is intended to be incorporated into $O_{int}(t)$; *topic* is the topic which the final user requests information about; $O_i(t)$ is the ontology whose root is *topic-according to-group i* in $O_{int}(t)$.

Add $O_j(t)$ to $O_{int}(t)$ as a mereological child concept, so that its root is *topic-according to-user j*
End

3.1.3 Select_Ontologies

Let candidates(t) be the set of candidate ontologies to be integrated. Let compatible$_i$(t) be the set of ontologies $O_j(t)$ belonging to candidates(t) that are compatible with $O_i(t)$.

For i=1 to Card(candidates(t))
 For i=j to Card(candidates(t))
  compatible$_i$(t)=compatible$_i$(t) ∪ O_j (t) if compatible($O_i(t)$, $O_j(t)$)
 End-For
End-For
Return the best subset according to the desired criterion (i.e., the subset with higher number of ontologies)
End
where

equivalent(x,y) is true if and only if for each concept belonging to x, there is another from y such that both of them have the same attributes and parent/children concepts and they are not temporally inconsistent concepts.
inconsistent(x,y) is true if and only if there are at least 2 concepts, one belonging to $O_i(t)$ and the other to $O_{int}(t)$, such that one of the following conditions holds:

a) They both have the same name, the concepts do not have any attribute in common and their respective parent/children concepts (if there were any) have the same attributes.
b) They both have the same attributes, there is no other concept, which is parent of one of them, with the same attributes than the attributes of any parent of the other concept. The same property holds for the children.
c) They are temporally inconsistent concepts.

compatible(x,y) is true if and only if (not(inconsistent(x,y) or equivalent(x,y)));
temporally inconsistent concepts(c(t),c'(t)) is true if there is a concept c''(t) which belongs to the same ontology as c(t), whose name is the same as the name of c'(t) and there is a concept c'''(t) which belongs to the same ontology as c'(t), whose name is the same as the name of c(t) such that one of the following conditions holds:

a) c(t) is a temporal parent concept of c''(t) and c'''(t) is a temporal parent concept of c'(t).
b) c(t) is a temporal child concept of c''(t) and c'''(t) is a temporal child concept of c'(t)

3.1.4 Ontological_Transformation

Let $O_{int}(t)$ be the integration-derived ontology and let $O_i(t)$ be each mereological child of $O_{int}(t)$ and n= number of mereological children of $O_{int}(t)$.

For i=1 to n
  For each concept c(t) belonging to $O_i(t)$ do
   If there is any concept c'(t) belonging to $O_{int}(t)$ such that
    equivalent_concepts(c(t),c'(t)) or (c(t) and c'(t) have the same name)
   then merge_attributes_and_relationships(c(t), c'(t))
   else link c(t) with its parents in $O_{int}(t)$
  End-for
End-for
where *equivalent_concepts(x,y)* is true if and only both concepts have the same attributes and parent/children concepts and they are not temporally inconsistent concepts.

4 GENERAL DESCRIPTION OF THE SYSTEM

The aim of the designed and implemented application was to develop a system and framework for managing a corporate memory that allowed an organization to take advantage of the knowledge supplied by the (internal or collaborating) groups belonging to the organization. The starting point of the system is a set of organization groups working in an intranet/internet and generating knowledge co-operatively but independently one group from another. In other words, this co-operation is totally transparent for each group because they do not know whether their knowledge is shared with other groups' one. A group is never allowed to see the knowledge created by another group directly nor modify other groups' work, but each group receives the global benefits from all the groups' (knowledge) contributions represented by ontologies. The system differentiates among two types of users, namely:

- *Group*: This is an organization working division unit, that is, a collective of people who generates knowledge for the system in such a way that other groups are able to look it up. Any group combine its own contribution with that of other groups of the same (co-operative) organization.
- *Manager*: This is the figure in charge of keeping the system working correctly. Another responsibility left to the manager is the management of groups as well as the knowledge to be maintained in the system.

A similar approach could be used for groups management. We could see each group as an organization, and we could split each group into two or more different types of users. We propose the following types of users in a specific group:

- *Employee*: This is a system worker, that is, a person who generates knowledge for the group(s) (s)he belongs to in such a way that other employees are able to look it up. Any employee may combine his/her own contribution with that of other employees from his/her same groups.
- *Administrator*: This is the figure in charge of managing the employees and the knowledge concerning a specific group.

4.1 Architecture and Implementation of the System

The solution we have adopted is to use a client/server architecture, where a group corresponds to a client and the corporate memory is kept in the server. Therefore, the software developed has two different parts, one for the client and another for the server. Knowledge integration is produced when an employee or group applies for it. It may happen that at that specific moment there exist some employees working on the generation of new knowledge they consider interesting for the organization such as new best practices or new versions of previous existing knowledge, so the knowledge of the corporate memory could have become obsolete. This represented a design problem we had to face, because there were two possibilities to choose: integrating the known system's knowledge until the moment the request is made, or actualizing the system's knowledge. We have decided to adopt in our prototype the second one because one of our goals was to maximize the quality of the information our users receive from the system.

However, this solution implies to keep track of the active users. Each time that an employee wants to have a look at the state of the corporate memory, the system checks a user register in order to know if its knowledge needs to be actualized. In case there is any possible new knowledge, it must be retrieved to increase the quality of the corporate memory. We needed to add new elements to our first architecture, so becoming more complex. Finally, the process of knowledge integration is briefly described as follows: Checking whether the knowledge in the corporate memory is up to date. If it is not, actualize it. Finally, supply the employee with the requested knowledge.

An employee can actualize the corporate memory, either because (s)he wishes to do it (by using the "Actualize corporate memory" option) or by an automatic actualization operation due to another employee's request. Once the knowledge has been integrated, the following step is to personalize the information. At this stage, the user has the chance for redefining the terminology that is assigned to the concepts belonging to the derived ontology. Then, the user will have better information about the topic than the one (s)he previously had when the request was made. Therefore, (s)he will be able to decide the terminology more accurately. Users are offered the possibility of changing the name that has been given to a concept by the process. The new name could be the one assigned by another user, who must have taken part in the integration process, or a different one that the user thinks to be more appropriate.

Attending to the properties and requisites that we have established in previous sections, a tool has been implemented in JAVA. Besides the client/server application, there is a web version that allows users to see the state of the corporate memory at a given instant.

5 A PRACTICAL EXAMPLE

The example we present in this section is based on an ontology built by last (fifth) year students at our university. The purpose was to build a co-operative enterprise and design a corporate memory model for it. We are not going to display the whole ontology but we are going to constraint our presentation to a general overview of the model, going (at first levels), in depth in the technical support area of the enterprise. The domain subject to study in this work was the film projectors industry. For it, several interviews with some domain experts were carried out before coming up with the model whose first levels are shown in Figure 3. This model has been got after integrating the (partial) models through the system we present in this paper. The complete model will be accessible at our group web page in the next months.

Figure 4 shows the part of the company that we have centered our efforts on, namely, the technical support department. We can see the knowledge schema about this department is divided into four parts: technical staff; the strategy of the department to face their working situations such as behavioral rules, working guidelines, etc; knowledge about the type of failures a projector can suffer from (diagnosis, treatments and best practices which help the technical staff to perform their job in a more efficient way); and suggestions about the company or the department.

In Figure 4, we can see the three different relationships between concepts: the knowledge about technical staff 'is a part of' the knowledge about the technical support department; the knowledge about adjustments 'is a class of' the knowledge about treatments of failures; the replacement of a bulb 'occurs after' a failure in the bulb has been diagnosed. The delay between the detection of the failure and the replacement of the bulb is represented as a fuzzy number, according to the temporal ontology representation described in Section 3. For example, the bulb is replaced between 5 and 10 minutes after the bulb failure has been diagnosed.

Finally, Figure 5 represents a screen snapshot of the system implemented. It is the part of the ontology that concerns the knowledge about the diagnosis process. We present here four possible families of reasons for project failure: sound, film, picture and bulb. We repeat that this model is not complete but only a brief introduction to what is feasible to do following our approach.

We can see the steps of corporate memory management in this example. We stated earlier in this section that we come up to this CM model after integrating different (partial) models. These (partial) models have been constructed by employees of the organization, in this case the students who simulated the (co-operative) organization. In this domain, employees are not supposed to know about ontologies or any other technology for representing knowledge. Therefore, the initial CM is constructed by the Knowledge Management department by using different techniques for extracting knowledge.

A group can be seen as a department in this organization and the administrator of each group can be a member of the Knowledge Management department because they are in charge of introducing the knowledge into the CM when an employee makes a request for adding new knowledge to it. This is part of the distribution of the CM whose description is continued next.

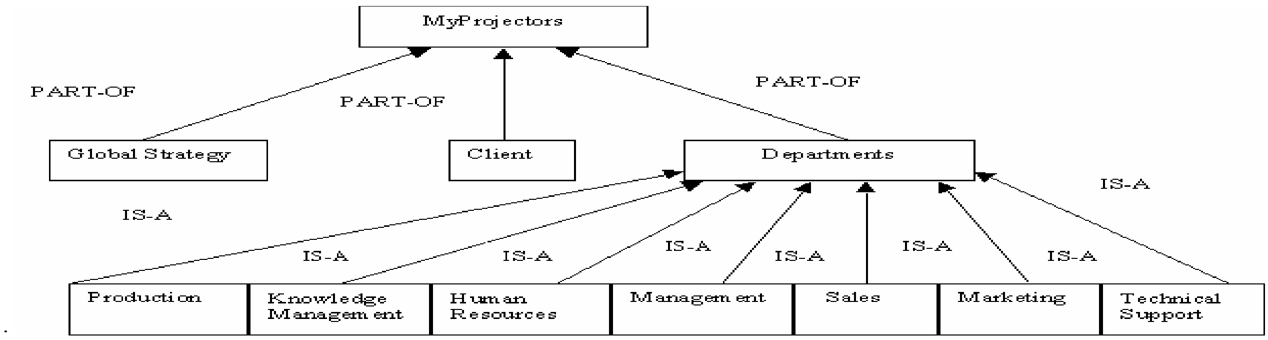


Figure 3. MyProjectors ontology

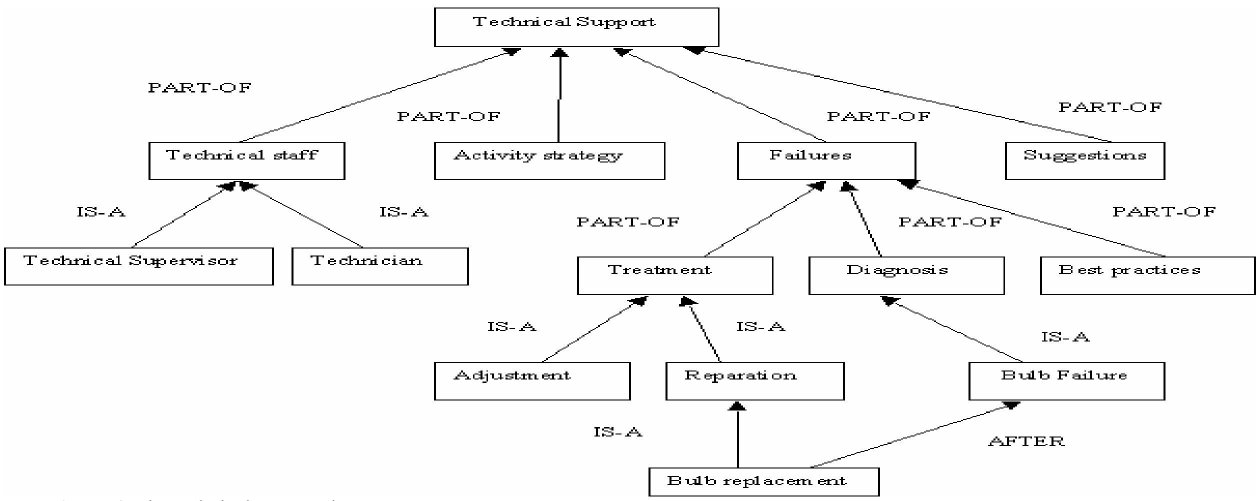


Figure 4. The technical support department

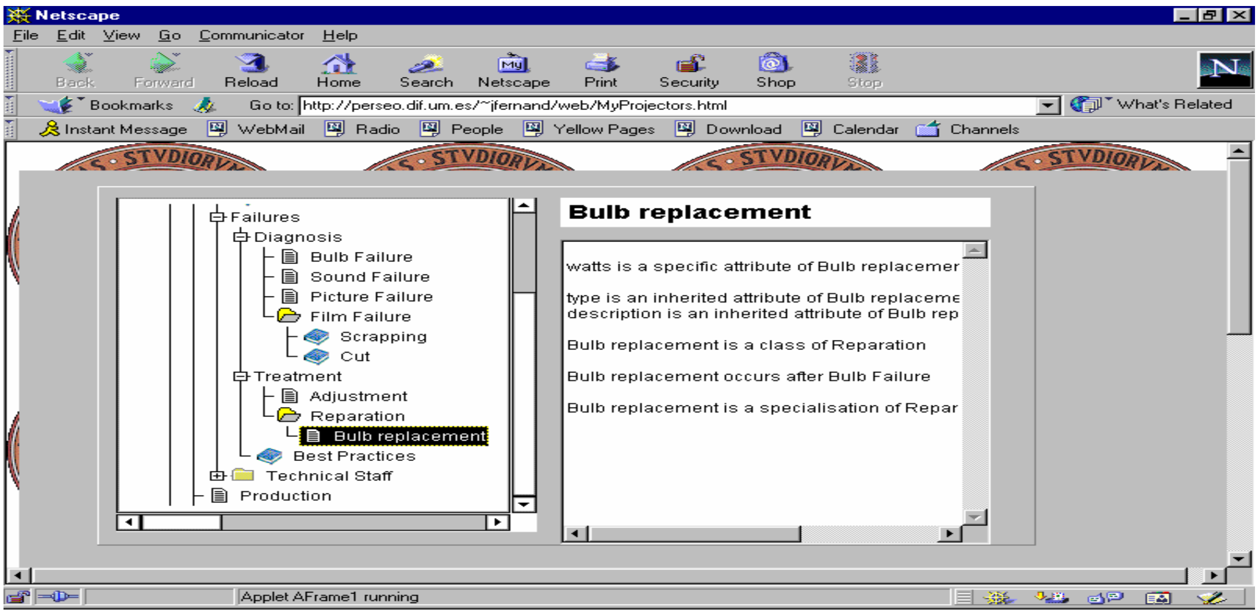


Figure 5. Using the tool for browsing the Corporate Memory

The model of the example represents a user request for checking the state of the CM. When this process is executed, the request goes to the administrator of his/her group, who is in charge of supplying the user with the best possible and accessible knowledge according to his/her preferences. Figure 5 shows the exploration of the concept 'Bulb replacement', and it represents a way of using the knowledge of the system for increasing the user's knowledge. The maintenance of the knowledge cannot be illustrated with this example because maintenance is a dynamic process while we are showing a snapshot of the system at a given instant.

6 CONCLUSIONS

Knowledge Management (KM) is an emergent topic in Artificial Intelligence and organizational environments. There is not any tool that provides a complete KM yet, but its current significance has encouraged the search for solutions capable of facilitating Knowledge Management. Thus, this work describes the design and implementation of a system through which employees of an organization can build, consult and maintain a Corporate Memory (CM) in a co-operative way. These (users) employees may be at different geographical locations (i.e. sites). The objective of a CM is to facilitate the sharing of the knowledge that exits within the organization in order to increase its productivity and competitiveness. The knowledge of employees is agreed to be the most important knowledge source for an organization but its main properties are its privacy and its tacit nature in most cases. The CM is an element that helps to make this knowledge public and explicit to other members of the organization. Our approach is similar to [32], where a CM is supposed to play two roles in the organization: passive (i.e., knowledge collector) and active (i.e., knowledge disseminator).

CM management implies to perform some key operations that have been detailed in this work. These include the construction of the CM from the knowledge which exists in the organization, its distribution to the staff of the organization, its use within the organizational frame, and the maintenance of the CM and the knowledge which is kept by the CM in order to ensure its correct temporal evolution. Some authors (see [31]) include information about external elements to the organization by splitting the CM into two: an external corporate memory and an internal one. Our approach is different since the way in which we construct the CM allows for the introduction of each piece of knowledge that is useful for the organization, independently of their (internal or external) origin.

Our approach covers the main processes in KM. The CM model represents an intellectual asset for the organization. In this approach, the generation of new knowledge is facilitated by the system, the corporate information is accessible and best practices can be shared by the members of the organization. Our CM model includes three different relationships (taxonomic, mereological and temporal), which allow employees to establish several kinds of relationships among the concepts they may be interested in. The definition of types of mechanisms for integrating knowledge facilitates one of the goals of a CM, namely, the reuse of knowledge to create new one in the context of organizations, so reducing the cost of obtaining it. This process is made through an adaptation of the terminology that is used for an employee's knowledge (formalized as an ontology) with respect to the global knowledge kept in the CM.

However, the selection of the most adequate terminology for the knowledge the user will receive depends on some parameters, particularly on two. The first one is the consistency of the knowledge that is intended to be introduced into the CM with the knowledge kept in the CM. The second parameter is the amount of knowledge which is contained in a specific piece of knowledge. An advantage of our approach is that the consistency of the knowledge of the CM is guaranteed due to the fact that each new (candidate) piece of knowledge to be included in the system is evaluated to check whether it is inconsistent with the current state of the system or not. In case there is any inconsistency between a new piece of knowledge of a user and his/her previous knowledge, the new piece of knowledge is considered to be the valid one and this one takes part of the integration process.

Collaborative knowledge building is not new. Thus, in [14], the authors have presented a system for collaborative construction of consensual knowledge bases. Such a system is based on the peer-reviewed journals: before introducing some piece of knowledge in a knowledge base, that piece must be submitted to and accepted by a given community. In order to achieve it, the definition of a protocol for submitting knowledge is provided. The consistency of the knowledge introduced into the CM is guaranteed by this principle and leads to the collaborative dialog among the experts. An important concern underlying this approach is that the community must use the same terminology. In our approach, a mechanism for synonym concepts management, that allows each agent to operate with its particular vocabulary, overcomes this problem. In order to solve the problem of synonym concepts, we use an approach close to that used by [33]. However, the way in which those conflicts are detected is different. In our approach, it is the system that is in charge of finding out which concepts are synonyms and which ones are not. This facility is not included in [33].

Three types of problems derived from the collaborative nature of the global ontology creation have been presented. These problems arise from the existence of different viewpoints different employees/groups can maintain with respect to a same domain. Then, the integration process can be seen as the unification of a group of consistent viewpoints. In literature, different approaches for managing inconsistencies among different viewpoints can be found. In [12], different consistency rules are defined to establish how the different viewpoints can be unified, and inconsistencies are prompted so that users choose how to solve conflicts. However, this solution cannot be applied in our approach because we intend to integrate automatically ontologies. In [9], a method for the comparison and integration of multiple viewpoints is presented. An extension of such approach can be found in [27] to represent and manage a Corporate Memory. In these works, different types of links are established among conceptual graphs. Moreover, different strategies, which can only be applied under certain circumstances, are used for solving conflicts. However, these links only cover relations between different entities, so that this approach does not take into account attributes. Therefore, this solution is not adequate for our approach.

Finally, some remarks about future work should be made. We plan to extend the approach in order to contemplate more (real) situations that can exist in an organizational context. For example, the inclusion of new types of relationships, extending the ones available now (i.e., taxonomic, mereological and temporal) is interesting because it will contribute to make our system more realistic and adequate to organizational environments. More facilities concerning users' preferences is another desirable future feature of the system. Another suggestion about future work is the inclusion of multimedia contents, which could make it easier to employees the understanding of specific concepts that belong to a domain in which they have not much background knowledge.

ACKNOWLEDGEMENTS

This paper has been possible thanks to the financial support of the Séneca Foundation, Coordination Centre for Research, through the Séneca Program(FPI). The research has been performed under projects FIT-150200-2001-320, FIT-070000-2001-785, and Seneca PL/3/FS/00.

REFERENCES

[1] Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., Sintek, M. Toward a Technology for Organizational Memories. *IEEE Intelligent Systems*, 1998, 13(3): 40-48.

[2] Abecker, A., Aitken, S., Schmalhofer, F., and Tschaitschain, B. KARATEKIT: Tools for the Knowledge –Creating Company. *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop*. Banff, Canada, 1998.

[3] Benjamins, V.R. & Fensel, D. (1998). Community is Knowledge! in $(KA)^2$. In *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, vol. 2, KM-2.1-KM-2.18,* Banff, Canada.

[4] Benjamins, V.R., Fensel, D. and Pérez-Gómez, A. Knowledge Management through Ontologies. *Proceedings of the 2nd Int.Conf On Practical Aspects of Knowledge Management (PAKM'98),* 1998.

[5] Borst, W.N. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD Thesis. University of Twente. Enschede, The Netherlands, 1997.

[6] Brooking, A. Corporate Memory: Strategies for Knowledge Management. International Thomson Business Press, 1999.

[7] Cardenas, M.A. A constraint-based logic model for representing and managing temporal information (in Spanish), PhD Thesis, University of Murcia, Spain, 1998.

[8] Caussanel, J., Chouraqui, E. Model and Methodology of Knowledge Capitalization for Small and Medium Enterprises. *Proceedings of Knowledge Acquisition Workshop'99*, 1999.

[9] Dieng, R. Comparison of Conceptual Graphs for Modelling Knowledge from Multiple Experts. In Z. W. Ras & Michalewicz eds., *Foundations of Intelligent Systems, Proc. of the 9th International Symposium on Methodologies for Intelligent Systems*, Zakopane, Pologne, June 9-13 1996, Springer-Verlag, LNAI n. 1079, p. 78-87.

[10] Dieng, R., Corby, O., Giboin, A., Ribière, M. Methods and Tools for Corporate Knowledge Management. In B.Gaines and M. Musen eds, *Proceedings of Knowledge Acquisition Workshop'98*, Banff, Alberta, Canada, April 18-23, 1998.

[11] Dignum, V., Heimannsfeld, K. Knowledge management for requirements engineering. *Procedings of Knowledge Acquisition Workshop (KAW'99)*, 1999.

[12] Easterbrook, S.M., Nuseibeh, B.A. Using ViewPoints for Inconsistency Management. *Software Engineering Journal*, Vol 11, No 1, Jan 1996.

[13] Eschenbach, C., and Heydrich, W. Classical mereology and restricted domains, *International Journal of Human-Computer Studies*, 1995, 43: 723-740.

[14] Euzenat, J. Corporate Memory through Cooperative Creation of Knowledge Based Systems and Hyper-Documents. In *Proceedings of Knowledge Acquisition Workshop*, 1996, p. 36-1 - 36-20, Banff, Canada.

[15] Fernández-Breis, J.T., & Martínez-Béjar, R. (2002) A cooperative framework for integrating ontologies. International Journal of Human-Computer Studies. To appear.

[16] Fridman-Noy, N. & Musen, M.A. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. 17th *National Conference on Artificial Intelligence, Austin, Texas.*

[17] Gómez-Pérez, A., Moreno, A., Pazos, J. and Sierra-Alonso, A. Knowledge Maps: An essential technique for conceptualisation, *Data & Knowledge Engineering,* 33(2), 169-190, (2000).

[18] Grunstein, M. La capitalisation des connaissances de l'enterprise, système de production de connaissances. *L'enterprise apprenante et les Sciences de la Complexité*. Aix-en-Provence, 1995.

[19] Martínez-Béjar, R., Benjamins, V.R. and Martín-Rubio, F. *Designing Operators for Constructing Domain Knowledge Ontologies*. In E. Plaza and R.Benjamins (Eds.), Knowledge Acquisition, Modelling and Management, Lecture Notes in Artificial Intelligence, 1997, 159-173, Springer-Verlag, Germany.

[20] McGuiness, D.L., Fikes, R., Rice, J., & Wilder, S. (2000). An environment for merging and testing large ontologies. In *Anthony Cohn, Fausto Giunchiglia, and Bart Selman, editors, KR2000: Principles of Knowledge Representation and Reasoning:483-493*, San Francisco, USA.

[21] Muller, H.J., Schappert, A. The Knowledge Factory – A Generic Knowledge Management Architecture. *Proceedings. of International Joint Conference on Artificial Intelligence*, 1999.

[22] Musen, Mark A. *Domain Ontologies in Software Engineering: Use of Protegé with the EON Architecture.* Report, 1997.

[23] Nagendra Prassad, M.V.N., Plaza, E. Corporate Memories as Distributed Case Libraries, *Proceedings of Knowledge Acquisition Workshop*, Banff, Alberta, Canada, November 9-14, 1996, p. 40-1 40-19.

[24] Nonaka, I., Takeuchi, H. *The Knowledge-Creating Company*. Oxford: Oxford University Press, 1995.

[25] O'Leary, D.E.. Using A.I. in Knowledge Management: Knowledge Bases and Ontologies. *IEEE Intelligent Systems*, May/June 1998, pp. 34-39

[26] Rabarijaona, A., Dieng, R., Corby, O. Building a XML-based Corporate Memory. *Proceedings of International Joint Conference on Artificial Intelligence*, 1999.

[27] Ribière, M. Using Viewpoints and CG for the Representation and Management of a corporate memory in Concurrent Engineering , in Proceedings of the 6th *International Conference on Conceptual Structures* ICCS'98, Montpellier, France, August 98

[28] Shaw, M.L.G. and Gaines, B.R.: A methodology for recognizing conflict, correspondence, consensus and contrast in a knowledge acquisition system. *Knowledge Acquisition 1(4) 341-363*, 1989.

[29] Steels, L. Corporate Knowledge Management. *Proceedings of International Symposium on the Management of Industrial and Corporate Knowledge*, 1993, Compiègne, France, pp 223-235.

[30] Stein, E. and Zwass, V. Actuallizing Organizational Memory with Information Technology. *Information Systems Research*, 6:2, 85-117, 1995.

[31] Van Heijst, G., Van der Spek, R., Kruizinga, E. Organizing Corporate Memories. In B.Gaines, M.Musen eds, *Proceedings of Knowledge Acquisition Workshop*, 1996, Banff, Canada, November, pp.42-1 42-17.

[32] Van Heijst, G., Schreiber, A. T., and Wielinga, B. J. Using explicit ontologies in KBS development, *International Journal of Human-Computer Studies*, 1997, 45: 183-292.

[33] Wiederhold, G. Interoperation, Mediation and Ontologies. *Proceedings. of Fifth Generation Computer Systems'94 Workshop on Heterogeneous Cooperative Knowledge-Bases*, Tokyo, Japan, p.33-48, December 1994.

To supply organization views, suited to users: an approach to the design of organizational memories

J.-Y. Fortier[1], **C. Cormier**[1], **G. Kassel**[1], **C. Barry**[2], **C. Irastorza**[1], **S. Bruaux**[1]

Abstract. Our work concerns the elaboration of organizational memories (OM). We investigate the feasibility and the benefit of a strong coupling between a knowledge base and a documents' base. Such a coupling supposes that the knowledge to manage is distributed, at one and the same time, in a knowledge model and documents. This distribution raises many questions such as: what knowledge to model?, and how to diffuse the modeled knowledge? In reply to the first question, we recommend modeling the organization for which the memory is elaborated, while insisting on the benefits and the genericity of the approach. For the diffusion of the modeled knowledge, we suggest introducing a mechanism of generation of documents adapted to the user's expectations. This paper presents our first results, in particular a generic software architecture which is currently developed. We illustrate these results with the elaboration of an OM for our research team, which constitutes a privileged experimental field for our work.

Key words: Methodologies and tools for Knowledge Management, Knowledge modeling and enterprise modeling, Semantic Portals, Ontologies and Information Sharing.

1 INTRODUCTION

In this paper, we present the first results of a project whose aim is to define a method and a software architecture to develop hybrid organizational memories (OM). Such OMs are based on the "strong" coupling between a knowledge base (KB) and a documents' base (DB).

The current approaches for the elaboration of hybrid memories consist in the coupling of an ontology with a DB to index documents and/or enable their annotation [2] [12]. These couplings are "weak" because the unique aim of the knowledge model – the ontology, and possibly, some annotations – is to facilitate the access to the documents. The knowledge which is managed is only in the documents. In our project, on the contrary, we study the feasibility and the benefit of a "strong" coupling. We consider that the knowledge to manage is both in the knowledge model and in the documents. We therefore seek to place the two forms of knowledge explicitation on the same level, searching for the strongest complementarity to manage knowledge.

Concerning the use of the knowledge, each explicitation form has indeed its own characteristics: the more the knowledge is structured, the easier it is to transform and to diffuse it; on the other hand, the more important is the initial cost for the formalization and the more complex is its maintenance. Then these different explicitation forms are complementary and they must be chosen by considering the value of the knowledge for the organization and its cost of acquisition and formalization [5].

The notion of documents' enrichment with the help of formal models of knowledge, as defined by [19] and exemplified in the project ScholOnto [6], is going towards a strong coupling. However, concerning the access to knowledge and its diffusion, an important dissymmetry persists. On the one hand, with the DB, we have many documents whose contents and form obey to an aim of communication on a fixed subject, for a fixed readership. On the other hand, with the formal knowledge model, we have a monolithic model whose role is not to inform about a fixed subject but to provide contextual knowledge to facilitate the access to documents and the interpretation of their contents. The knowledge model is then always oriented towards the documents and its contents don't benefit of the same facilities of diffusion.

In order to realize a true strong coupling between a KB and a DB and to provide the modeled knowledge with the same facilities of diffusion, we suggest to introduce a mechanism of generation of documents from the modeled knowledge. In response to an user's request concerning a theme tackled in the knowledge model, a structured description, whose contents are adapted to the expectation of the user, is constructed. This mechanism is integrated in a larger document, that we call "knowledge book", playing the role of portal for the memory [24]. The knowledge book presents the subjects tackled in the knowledge model and assembles, for a fixed user, a set of predefined requests.

In order to define our approach and evaluate our software architecture, we have chosen to manage the knowledge of our

[1] LaRIA, Knowledge Engineering team, University of Picardie Jules Verne 5, rue du Moulin Neuf – 80000 Amiens, France, <name>@laria.u-picardie.fr

[2] PSI, INSA of Rouen, Place Émile Blondel, BP 08, 76131 Mont Saint Aignan Cedex, France greboval@insa-rouen.fr

research team [16]. We have also developed a prototype of OM, in a near context of industrial research [11].

In the paper, we complete the description of our work according to the following plan:

- Section 2 describes the principles followed to achieve a strong coupling between the knowledge model and the documents.
- Section 3 presents a generic software architecture of OM, whose components are currently implemented. In particular, we see how the module in charge of the construction of documents, called "Writer", collaborates with the modules implementing our knowledge representation language DefOnto [17]: a compiler and an inference engine.
- Section 4 presents the method of ontologies construction, OntoSpec [15], which is associated with the approach, and the role played by the two manifestations of the ontology that the method considers: a semi-informal ontology and a computational ontology specified in DefOnto.
- Section 5 emphasizes the contribution of DefOnto in the elaboration of the memory, in terms of power of expression and inferential services.
- Section 6 presents the prototype of OM realized in the project of managing our team knowledge.
- Section 7 assembles comparisons between our work and other projects.

2 PRINCIPLES OF A STRONG COUPLING

This section presents principles retained to realize a strong coupling. They concern the choice of knowledge to model §2.1 and the way the modeled knowledge is diffused §2.2.

2.1 To model the organization: interest and genericity of the approach

As the choice of the knowledge to model depends, as we said before, on many parameters, notably on the value of knowledge for the organization, one has to expect that this choice varies from one application to the other. In this section, we present the choice that we have undertaken in our projects. In addition we put forward arguments to consider this choice as generic, that is to say susceptible to be retained in a large number of applications.

We suggest modeling the organization for which the memory is constructed and to elaborate this model by successive refinements.

A model of the organization describes the structure of the organization, its members, its partners, its activities, its products or results, and its documentation. Such a model satisfies different objectives, often put forward to motivate projects of OM construction:

- *To help diffuse the documentation*. The model contains many references to documents. For example, the description of actors or partners of the organization makes reference to documents of which they are authors or publishers. Similarly, the description of activities led within the organization (e.g, meetings, projects) makes reference to related documents (e.g., meeting reviews, documentation of project). Therefore the model of the organization offers a context to access to documents.
- *To help integrate a new actor in the organization.* Such a model notably informs the newcomer of activities to which the different members of the organization participate, their responsibilities in these activities, the state of project furtherance. In addition, the newcomer is aware of documents that he/she has to consult as a priority and of their location.

As we can see, a simple model of the organization, making no reference to the business knowledge of the organization, can offer a good return on the investment in time required for modeling knowledge. It allows to construct, at a low cost, a first version of the memory and to interest actors of the organization in the knowledge management project.

Thereafter, such a model can be complicated, notably by modeling business knowledge of the organization. It becomes then possible, for example, to describe the competence of organization members by connecting them to business objects (e.g.: [7]), to preserve knowledge for activities judged crucial (e.g.: [13]) or to index documents by their contents to facilitate their access (e.g.: [22]). We thus recommend an incremental development of the model of the organization, each supplementary effort of modeling leading to the addition of a new service to the memory.

Should be noted that elaborating the model of the organization supposes that the notions necessary for its expression have been beforehand defined, for example the notions of "partner", "confidential document" or "internal project". Here we meet the question of the construction of ontologies. It is necessary therefore to consider that the KB contains both the model of the organization and its associated ontology. In section 4, we deal with the method of construction of the ontology and we emphasize the role played by the ontology in the exploitation of the memory.

2.2 Diffusion of the modeled knowledge by means of documents suited to the profile of the user

The KB is specified in a formal language of representation, as far as we want the OM to reason about the modeled knowledge, and we therefore face the question of the mode of diffusion of this knowledge.

The contents of the KB can't be indeed diffused as they stand, for several reasons. On the one hand, representations of knowledge are specified in a computer language and are therefore hardly understandable for a human being. On the other hand, the order of the entities of representation in the KB corresponds to a KB development logic, not to a knowledge transmission logic. In addition, it is out of the question to present the totality of the KB to a member of the organization, if only because of access rights to some information.

We therefore suggest generating from the KB a "readable" document, whose contents are adapted to the needs of the user, that we call "knowledge book". Such a document contains a summary (a table of contents) gathering, in the way of a classic book, an ordered set of themes. Each theme corresponds to a subject tackled in the knowledge model, in our case to a partial view of the organization. More precisely, a theme corresponds to an object, or a list of objects, described in the knowledge model, for example a person, a project or an organization. The selection of a theme in the summary, to consult the contents of the corresponding section of the book, leads to the construction on the fly of the presentation of the corresponding object (or set of objects), by a software module called "Writer". The Writer takes into account both predefined models of presentation of types of objects (one does not present a person like a project), the ontology and the profile of the user, to elaborate a structured description.

Thus, each user receives a book, where both the summary and the contents of sections are adapted to his expectations and to his profile. This supposes that the memory disposes of different summaries, adapted to the different profiles of users. These summaries are defined by a member of the organization playing the role of administrator of the memory.

3 SOFTWARE ARCHITECTURE

Let's see how the different functionalities we have just described are materialized in term of a software architecture. Figure 1 shows a general view of our OM architecture (the main software modules are graphically represented by rectangles). It defines two different roles that may be played by organization actors, the roles of editor and consultant.

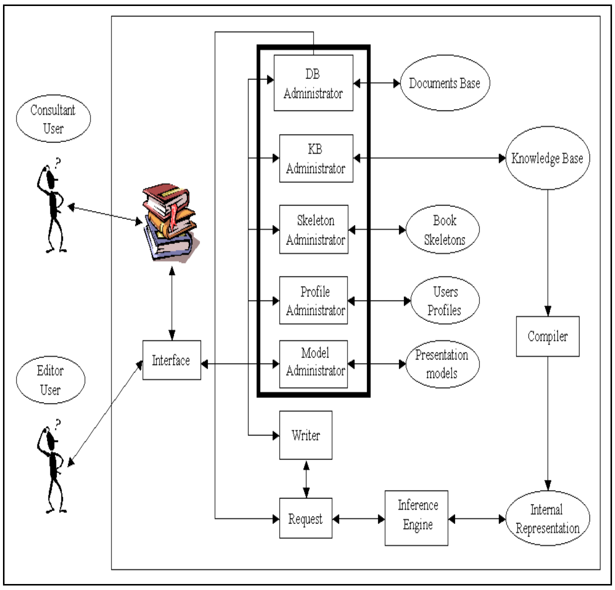


Fig. 1. OM architecture

Acting as an editor, the user edits different knowledge sources in order to build them up and maintain them:

- The Knowledge Base, composed of the ontology and the organization' model, is formalized in the DefOnto language. This KB is translated by a compiler to an internal representation (in JAVA), whose structure is optimized to allow inferences. The addition of new descriptions in the organization' model leads to extensions of the internal representation (incremental compilation). The inferences are realized by the Inference Engine.
- The Documents' Base. The addition and the deletion of documents are assumed by the document administrator, who also maintains the DefOnto description of the documents in the KB, with the help of the KB Administrator. We can notice that some documents can be described in the KB without being stored in the DB. This is the case, for instance, with papers, archived in cupboards, or with web sites for which we only archive the address.
- The user profiles, which consist in a list of user "types": consultant (internal or external to the organization), editor (of the ontology, of the organization model, etc.). Each type of user is linked to a book skeleton and some rights of access regarding the level of confidentiality of the knowledge or the documents.
- The book skeletons, which contain the structured summaries of the books. A summary is defined by a list of themes and sub-themes corresponding to partial views of the model of the organization. Each terminal element of the summary corresponds to a precise information request, related to one object or a list of objects, that is sent to the Writer by the Interface.
- The presentation models, which consist in an ordered set of properties for a given type of object. Yet for a same type of object to present, the model may differ according to the rights of the consultant (an external person can't see confidential information), the level of description expected (we don't expect the same description in an introduction and in a sub-section dedicated to this object).

As we can see, there are different types of editors, each one needing some special abilities: the editor of users profiles must be aware of consultants needs whereas the ontology editor should be a knowledge engineer with knowledge modeling skills and so, he may not be an employee of the organization.

Acting as a consultant, the user has access to his knowledge book. At that time, he is able to perform different actions:

- To select an entry of the summary, that leads to the writing of the corresponding section by the Writer and its visualization by the Interface. The related information request is translated by the Writer to elementary requests that are transmitted to the Request module. The latter searches into the KB using the Inference Engine to answer these elementary requests. In the end, the Writer exploits the produced answers, the user profile and presentation models predefined for each type of entity (e.g.: person, project) to write the section.
- To express an information request, not anticipated in the summary, and whose answer will flesh out the book. The user has to deal with the Writer, which will help him to express his request. The rest of the processing is similar to that of a predefined request of the summary.

4 BUILDING AND ROLES PLAYED BY AN ONTOLOGY IN THE CONCEPTION AND EXPLOITATION OF THE MEMORY

We have seen that the KB is composed of a model of the organization and of an ontology. The latter has different functions, constituting an help for the conception and exploitation of the model of the organization. In this section, we emphasized the contribution of the ontology to strong coupling, from methodological point of view, successively presenting: the method "OntoSpec" for building ontologies [15], which is integrated in our method of OM building §4.1; the two manifestations of the ontology considered by OntoSpec, a semi-informal conceptual ontology §4.2 and a computational ontology §4.3, specifying their respective role; the ontology OntoOrg, built in different projects of memories construction, and which constitutes a resource bound to our method §4.4.

4.1 Ontology construction

The method OntoSpec [15] results from an evolution of the methodological framework defined in [18]. OntoSpec suggests to organize the development of an ontology with two main steps named "ontologization" and "operationalization":

- *Ontologization* corresponds to acquisition and modeling of ontological knowledge (the notions). It is guided by modeling primitives (e.g., concept, relation, essential property), the specification being made at the "knowledge level", which means that no computer constraint is taken into account (e.g., language syntax, inference time). This step leads to a *conceptual ontology*, specified in a semi-informal way.
- *Operationalization* takes as data the *conceptual ontology* to code it into the language of representation DefOnto. As DefOnto is also a programming language, this step leads to a *computational ontology.*

Such a decomposition is inherited from KBS building methods which, like the CommonKADS methodology [23], distinguish two levels of modeling: a modeling to make sense and a modeling to implement a system. We also find it in methodologies for ontology building like METHONTOLOGY [10] and TERMINAE [4][3].

Among these two steps, ontologization is certainly the more crucial step. It allows cooperative work between a knowledge engineer and the actors of the organization receiving the memory, to get a coherent, complete and consensual, system of concepts. It is led by the tasks that the memory must assist, tasks which determine the nature of the organization' model to consider. Once the conceptual ontology obtained, the operationalization consists in coding the modeled knowledge using the language DefOnto. This step can be done by a computer scientist who knows the constructions of the language and its inferential services.

4.2 The conceptual ontology

The conceptual ontology is specified in a semi-informal way, which means that definitions of conceptual entities (concepts and relations) are expressed in a strongly structured and controlled natural language.

The structure of a definition (*cf.* examples in figure 2) is based on a classification of propositions which are likely to contribute to the contents of the definition:

- Some propositions are used to express properties of objects denoted by the conceptual entity. At a first level, the properties are classified as "essential" properties (EP) or "incidental" (IP)[4]. At a second level, the properties are classified according to "roles" they play regarding the defined conceptual entity. These roles can be abstract (e.g., Necessary Condition (NC), Sufficient Condition (SC)) or more specific, and in this case specialize the previous (e.g., Subsumption Link (SL), Subsumption Link with Differentia (SLD), Link of Mutual Exclusion (LME), Relational Link (RL), Domain Restriction (DR), Range Restriction (RR)).
- Other propositions are used to express comments, aiming, either at clarifying the definition supplying examples and/or counter examples, or, for the modeler, at memorizing choices of modeling. An important example of comment, intended to reinforce the understanding of definitions, consists in explaining the presence of "semantic axes" (SA) when a notion is specialized according to several dimensions. So, the notion of "document" can give birth to notions of "electronic document" and "paper document", according to the physical support used, notions of "document in French" and "document in English", according to the language used, finally notions of "announcement of thesis presentation" and of "call to participation to scientific event", according to the communicating intention of the document's author.

Employee: [EP/SLD] an EMPLOYEE is a PERSON who WORKS ON BEHALF OF an EMPLOYER. [EP/RL] Every EMPLOYEE IS PAID BY the EMPLOYER who employs him. [SA] The concept EMPLOYEE is specialized in ENGINEER, RESEARCHER according to the nature of work realized by the EMPLOYEE.

Electronic document: [EP/SLD] An ELECTRONIC DOCUMENT is a DOCUMENT which HAS A SUPPORT electronic. [EP/RL] Every ELECTRONIC DOCUMENT HAS FOR FORMAT a FORMAT. [EP/LME] The ELECTRONIC DOCUMENTS are opposed to PAPER DOCUMENTS.

Works on behalf of; is employed by: [EP/SL] WORK ON BEHALF OF implies TO BE USED AS A RESOURCE BY. [DR & RR].An EMPLOYEE WORKS ON BEHALF OF an EMPLOYER.

Fig. 2. Semi-informal definitions of two concepts and of one relation

In addition to this catalogue of propositions, whose role is to control the contents of a definition, the knowledge engineer has rules at his disposal to control the expression in language, such as:

- Rules to paraphrase each type of proposition, in order to get homogenous definitions.
- Typographical conventions to place words used in the definition in relation to the current conceptualization. For ex-

[3] A comparison of OntoSpec with these methods is out of range of this paper. The interested reader will read (Kassel, 2002).

[4] The essential properties are verified by the objects denoted by the concept in every situation, or world, possible. They are thus "really" definitional. Conversely, the incidental properties are satisfied only in the subrange of situations where memory is likely to be confronted.

ample, when a term is used in a meaning corresponding to a notion of the ontology, this word is written in capital letters.

In addition to being used as a basis for coding the computational ontology in DefOnto, the conceptual ontology becomes encapsulated, as it is, in the computational ontology, which allows it to be exploited by the Writer. So, to answer a query on the sense of a term, the Writer exploits the semi-informal structured definition associated to the concept, to extract the definitional properties and to suggest a definition of the term.

4.3 The computational ontology

The computational ontology is specified with the language DefOnto [17]. It is obtained by coding semi-informal propositions into formal propositions (*cf*. figure 3). However, due to a limited propositional power of expression of DefOnto, some semi-informal propositions don't have their equivalent in DefOnto. In the definition of the concept EMPLOYEE, it is for example the case for the second proposition[5]. This reduction of sense justifies that we keep the conceptual ontology encapsulated in the computational ontology.

DefOnto is a compiled language. The formal ontology, and declarations of objects which are instances of the concepts of ontology, are translated into an internal data structure (*cf.* figure 1). The compiler successively makes a lexical and syntactic analysis, then a semantic one of the internal representation of DefOnto. The internal data structure is optimized to provide inferential services. The latter are described in §5.2.

```
(DefGenConcept #employee
    = [#person] -> (ME#works_on_behalf_of) -> [#employer])
(DefGenConcept #electronic_document
    = [#document] -> (MI#has_a_support) -> "electronic"
    ObjectsProperties
        -> (AE#has_for_format) -> [#format]
    SetProperties
        -> (#is_disjoint_with) -> [#paper_document])
(DefRelation #works_on_behalf_of
    IsA [#is_used_as_resource_by]
    RelationProperties
        -> (#has_for_domain) -> [#employee]
        -> (#has_for_range) -> [#employer])
```

Fig. 3. Definition in DefOnto of two generic concepts and one relation

[5] This proposition "every employee is paid by the employer who employs him" has for equivalent in first order logic : *"x"y employee(x) ® (employer(y) Ù employs(y,x) ® ispaidby(x,y))*. The use of the variable y, in logic, allows to bind the entity which employs to the one which pays. The lack of variable in DefOnto explains why we can't represent this proposition.

4.4 OntoOrg: an ontology dedicated to the management of organizational knowledge

The experience that we have accumulated in different projects of memory building shows that the construction of the ontology remains, in spite of the existence of methodological guides, a complex process which constitutes a real bottle neck for the step of knowledge modeling. This fact explains why we consider the reuse of existing ontologies as a critical aspect for the process of ontology building. In our project, we approach this aspect from the point of view of the management of the life-cycle of ontologies developed for different applications.

To build the ontology OntoPME, within the framework of our project of OM for our team [18] [16], we mainly reused the ontology of the project (KA)[2] [2] and, to a lesser extent, Enterprise Ontology [25]. Recently, for a second project, we reused OntoPME to build OntoDCRIT by adapting OntoPME to the needs of a new organization [11]. A work in progress consists in integrating the two ontologies into a generic ontology OntoOrg, which builds on the needs met in the two projects, while erasing the particularities of the concerned organizations. The stake of this work is to have a resource with growing quality to reduce the cost of ontology building for future applications.

OntoOrg ontology is composed of five sub-ontologies corresponding to five great themes, or types of objects: activities, documents, events, organizations and persons. Figure 4 graphically shows semantic axes structuring the sub-ontology of documents[6].

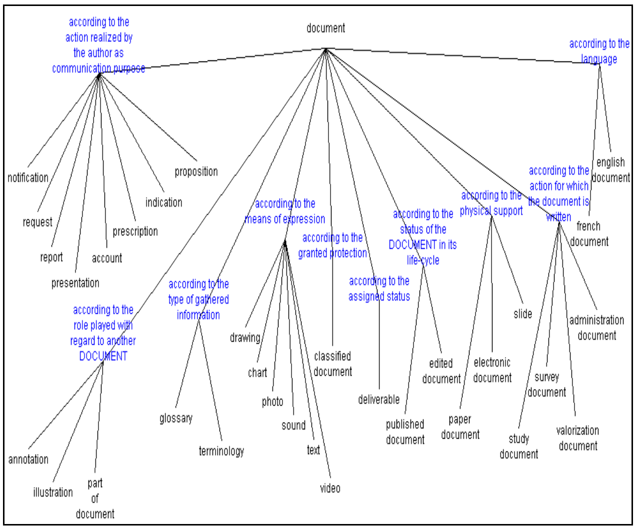


Fig. 4. Different specializations of the notion: document

5 CONTRIBUTION OF THE LANGUAGE DefOnto TO STRONG COUPLING

In this section we go back to DefOnto to put ahead two important aspects of the language regarding our goal of strong cou-

[6] A french version of OntoPME can be consulted at URL : http://www.laria.u-picardie.fr/EQUIPES/ic/demo/onto-pme.html. OntoOrg will be on-line on July 2002.

pling. On the one hand, we present the power of expression offered by DefOnto to formalize the model of the organization §5.1. On the other hand, we describe the inferential services provided by DefOnto and the query language used to make the coupling between the Inference engine and the Writer §5.2.

5.1 Formalization in DefOnto of the model of the organization

In addition to the representation of generic concepts, or classes of objects, DefOnto allows to represent individual concepts, *i.e.* points of view on individual objects, and this capacity is used to formalize the model of the organization. Two original characteristics, which are not shared by other languages [17][7], confer to DefOnto a great power of expression for this purpose. They are illustrated in the representations presented in figure 5.

A first characteristic is the possibility to define relations on relations, which confers to DefOnto a large propositional power of expression. This possibility is used in the description of the object *#KE_team* to represent the complex proposition: “KE team takes part in A2C2 project with HEUDIASYC partner, since January 1^{st} 2002”.

A second important characteristic is the possibility to define meta-knowledge, which allows the definition of classes of concepts, propositions and entities of representation. It becomes thus possible to represent the following knowledge: “the fact that the KE team takes part in mounting XX007 project is a confidential information” (the concept *#confidential_information* is defined as a class of propositions); “The entity representing the document (Fortier, 2001) was put in the KB on October 5^{th} 2001” (the property *#has_for_intrance_date_in_KB* bears on the entity of representation and not on the object).

We have just seen with these examples that DefOnto allows to represent a relatively complex model of the organization, that is assuredly an important point regarding our goal to manage knowledge at once in the knowledge model and in the documents.

[7] Comparisons of DefOnto with other languages of representation (e.g., LOOM and OIL) are gathered on site: http://www.laria.u-picardie.fr/EQUIPES/ic/LangComp/

```
(DefIndConcept #KE_team
   IsA [#research_team]
   ObjectProperties
      -> (#is_a_component_of) -> [#LaRIA]
      -> (#has_for_responsible) -> [#Gilles_Kassel]
      -> (#takes_part_in_project) –
         -> [#A2C2]
         -> (#with_partner) -> [#Heudiasyc]
         -> (#since) -> [#january_1st_2002],
      -> (#takes_part_in_mounting_project) –
         -> [#XX007]
         -> (#proposition_belongs_to) -> [#confidential_information],)
(DefIndConcept #(Fortier, 2001)
   IsA {[#LaRIA_internal_report], [#stage_report]}
   ObjectProperties
      -> (#has_for_publishing_date) -> [#september-1st-2001]
   EntityProperties
      -> (#has_for_intrance_date_in_KB) -> [#october-5th-2001])
```

Fig. 5. Definition of two individual concepts in DefOnto

5.2 Inference services of DefOnto

DefOnto provides a query language constituted of a range of filters types. Each filter type corresponds to a particular type of query bearing on contents of the KB, notably:

– To return the explicit extension of a concept of the ontology, for example to return all internal reports: [#internal_report *x].
– To compute the extension of a concept of the ontology taking into account the ontological knowledge, for example to determine all the internal reports taking into account the fact that an activity report is an internal report: (can-infer-than [#internal_report *x]).
– To determine if an object explicitly (resp. implicitly) belongs to the extension of a concept of the ontology, for example to determine if (Cormier *& al.*, 2002) is an internal report: [#internal_report #(Cormier & al., 2002)], or (can-infer-than [#internal_report #(Cormier & al., 2002)]).
– To determine the set of linked objects to a given object according to a given relation, for example to determine who are the authors of (Cormier & al., 2002): [#has_for_author #(Cormier & al., 2002) *y].

Theses queries are transmitted to the Query module by the Writer and are evaluated by the Inference engine. A request of the user, for example, find all internal reports published from a given date, can correspond to a conjunction of filters. The role of the Query module is also to integrate results of the evaluation of different filters.

6 REALIZATION OF AN OM PROTOTYPE

Within the framework of our project PME (project of team memory), we have developed an OM prototype. This one is composed of two knowledge books using the same KB. A first book, accessible on Internet[8], presents the KE team of LaRIA and more widely the KE community in France and abroad by presenting teams, projects and documents, of reference. A second book, only accessible on a team's intranet, constitutes a work tool for the team. In addition to the information available in the first book, it permits to edit documents with restricted diffusion (work notes, reports of meetings, etc.) and indicates more detailed information on the team's projects.

Such a book (*cf.* figure 6) consists in two parts: the left part corresponds to the visualization of a table of contents and the right part corresponds to the visualization of the contents of the sections. The latter corresponds to a partial view of the organizational model generated by the Writer.

The table of contents is made up of a set of ordered themes. For example, the editor of the book (accessible on the Internet) has estimated that the presentation of the KE team of LaRIA should begin with the general presentation of the team and should continue with a presentation of its members, then of its partners. The themes can be broken down in sub-themes. The theme "Its projects", in our example, is broken down in two sub-themes: "internal projects" and "projects in collaboration".

A navigation in the table of contents allows the user to select a theme. The activation of the theme generates the construction of the corresponding view. This view corresponds to the structured description of objects[9].

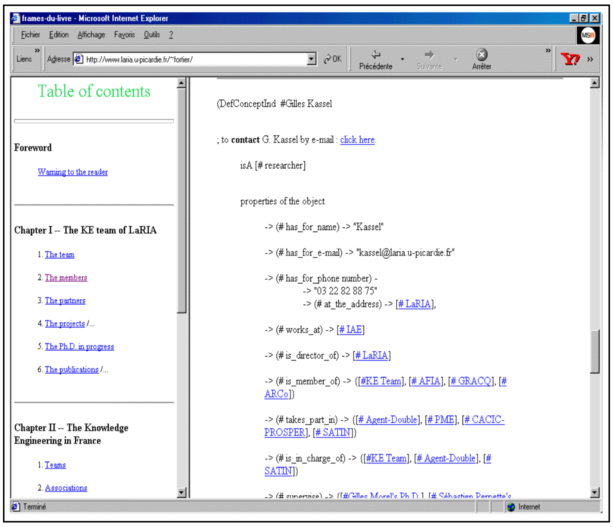


Fig. 6. Visualization of the knowledge book

In our example (*cf.* figure 6), the contents in the right window correspond to the results of the activation of the theme "its members". After the Writer retrieved all the necessary elements, he sends a request to the Request component to obtain the objects and the properties to show to the consultant. Then, he writes a XML file with these results and sends it to the Interface which can easily process it to adapt the order of presentation. This possibility is interesting only for a list of objects. The interface can change the order of presentation according to the user' choice. In our example, the Interface shows a response to the consultant which consists in an ordered list of descriptions of the members of the team; it begins with the presentation of the team leader. Then the permanent members are presented. The Ph.D. students and the associated members terminate the presentation. For each member, his/her name and his/her address are first indicated, then the projects in which he/she participates, and, if they exist, his/her responsibilities. The consultant may prefer a presentation according to the project in which the person participates rather than according to the administrative function. Concerning documents, the consultant can prefer to order the presentation by date, by author or by subject.

The objects' descriptions mention different entities: concepts and relations which are part of the ontology (e.g.: researcher, supervisor), and other objects which structure the model of the organization. Links on these entities (*cf.* figure 6) allow the user to get other knowledge. They allow to see a definition of concept or relation, or the description of another object. When following the links, the user can in particular reach the description of the set of referenced documents.

Finally, some actions are allowed on certain objects with the purpose of interacting with the entities of the physical world that the objects model. For example, it is possible to edit documents for which we have an electronic version with the help of the document administrator which is in charge of maintaining the document' base or to contact someone with the mail. In our example in figure 5, we can contact Gilles Kassel in activating the link "click here".

7 RELATED WORKS

In this section we compare our memory architecture to other architectures relying on a KB and DB coupling.

We find in the (KA)² project [2] and its recent continuations [24] a memory architecture close to ours: the KB is made with web pages annotations, and the ontology is used both to model annotations and to infer implicit knowledge during the queries. The replies consist however in objects lists, not in structured objects descriptions. Moreover, these replies don't take a user profile into consideration.

The CoMMA European project [12] mainly aims at evaluating the contribution of a multi-agents approach to design and to implement OM. It exploits emergent web technologies (XML, RDF(S)) for the annotations and ontology specification. With the translation of these RDF(S) specifications to conceptual graphs and the use of CORESE inference engine [8], we find again an architecture close to ours. Moreover this project has recently led to an expansion of RDF(S) to extend the expression capability for the ontology and annotations specification [9]. Nevertheless, as in the (KA)² project, the query replies only consist in elementary objects lists.

[8] http://www.laria.u-picardie.fr/EQUIPES/ic/demo/livre-ic.html

[9] Actually, the prototype that implements our software architecture directly uses the files which compose the KB formalized in DefOnto. A new version integrating the functionalities of the "writer" module will be soon accessible.

In the ScholOnto and myPlanet projects [6][14], which rely on the notion of documents enrichment [19], the KB contains knowledge to facilitate the documents access and their contents interpretation. The KB/DB coupling is therefore used, as in $(KA)^2$ and CoMMA, to make the information retrieval easier by using knowledge models.

In our approach, the model of the organization plays the same role of documentation contextualization, but it is besides exploited for itself, in a strong coupling perspective. This exploitation goes through the addition of a diffusion mechanism for the modeled knowledge, which takes the form of a generation of customized virtual documents.

8 PROSPECTS

The works we have just exposed are going into different directions.

A first version of the set of software modules, which constitutes our OM architecture, is currently being built. The multi-agents approach, already used in different OM projects [1] [12], seems well suited for the implementation of such software architectures. We have chosen to use the JADE platform [3] as in the CoMMA project.

Currently, the presentation models are linked to the objects in the book skeleton, which leads to duplicate these models and also to incorporate the user' profiles in these models. To overcome these limitations, we plan to adopt a knowledge-based approach for the Writer which will dispose of generation methods for the elaboration of structured descriptions. Such an approach will provide us more flexibility to take into account the users profiles.

At the same time we plan to carry out other experiments and capitalize on the experience both in the software architecture and in the associated OM development method.

9 REFERENCES

[1] BAEK S., LIEBOWITZ J., PRASAD S.Y., GRANGIER M. (1999). Intelligent Agents for Knowledge Management - Towards Intelligent Web-Based Collaboration within Virtual Teams. In J. LIEBOWITZ Eds., Knowledge Management Handbook, CRC Press, p. 11-1, 11-23.

[2] BENJAMINS V.R., FENSEL, D., DECKER, S., GÓMEZ-PÉREZ A. (1999). $(KA)^2$: Building Ontologies for the Internet: a Mid Term Report. In *Proceedings of the International Workshop on ontological Engineering on the Global Information Infrastructure*, Dagsthul Castle (Germany), p. 1-23.

[3] BERGENTI F., POGGI A., RIMASSA G. (2000). Agent Architecture and Interaction Protocols for Corporate Memory Management Systems. In *Proceedings of the ECAI'2000 Workshop: Knowledge Management and Organizational Memories*, p. 39-47.

[4] BIÉBOW B., SZULMAN S. (1999). TERMINAE: A linguistic-based tool for the building of a domain ontology. In *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management: EKAW'99*, Dagstuhl (Germany).

[5] BUCKINGHAM SHUM S. (1997). Balancing Formality with Informality: User-Centred Requirements for Knowledge Management Techniques. In *Proceedings of the AAAI Spring Symposium Workshop "AI in Knowledge Management"*, Stanford University, p. 127-133.

[6] BUCKINGHAM SHUM S., MOTTA E., DOMINGUE J. (2000) ScholOnto: An Ontology-Based Digital Library Server for Research Documents and Discourse. *International Journal on Digital Libraries*, 3(3), p. 237-248.

[7] CLARK P., THOMPSON J., HOLMBACK H., DUNCAN L. (2000). Exploiting a Thesaurus-Based Semantic Net for Knowledge-Based Search. In *Proceedings of the Twelfth Conference on Innovative Applications of Artificial Intelligence: IAAI-2000*, Austin (Texas, USAA), p. 988-995.

[8] CORBY O., DIENG R., HEBERT C. (2000). A conceptual Graph Model for W3C Resource Description Framework. In *Proceedings of the 8th International Conference on Conceptual Structures (ICCS'2000)*, August 2000, Darmstadt, Germany.

[9] DELTEIL A., FARON C., DIENG R. (2001). Extension of RDFS based on the CG formalism. *To appear in the Proceedings of ICCS'01*, August 2001, Stanford (USA), p. 275.

[10] FERNÁNDEZ-LÓPEZ M., GÓMEZ-PÉREZ A., PAZOS-SIERRA J., PAZOS-SIERRA A. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, January/February, p. 37-46.

[11] FORTIER J.-Y. (2001). Construction d'une ontologie pour gérer la documentation d'un centre de recherche. Application au Centre Commun de Recherche d'EADS. Internal report 2001-08 of the LaRIA, University of Picardie Jules Verne, October 2001. *Title in english: Design of an ontology for managing the documentation of a research center - Application to the Common Research Center of EADS.*

[12] GANDON F., DIENG R., CORBY O., GIBOIN A. (2000). A Multi-Agent System to Support Exploiting an XML-based Corporate Memory. In *Proceedings of the 3rd International Conference on Practical Aspect of Knowledge Management (PAKM'2000)*, p. 10-1, 10-12.

[13] GOLEBIOWSKA J., DIENG-KUNTZ R., CORBY O., MOUSSEAU D. (2001). Exploitation of Ontologies for Building A Vehicule Project Memory. In *Proceedings of the IJCAI'01 Workshop on Knowledge Management and Organizational Memories*, Seattle (USA), p. 49-58.

[14] KALFOGLOU Y., DOMINGUE J., MOTTA E., VARGAS-VERA M., BUCKINGHAM SHUM S. (2001). *myPlanet*: an ontology-driven Web-based personalised news service. In *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, Seattle (USA), p. 44-52.

[15] KASSEL G. (2002). OntoSpec : une méthode de spécification semi-informelle d'ontologies. In Actes des Journées Francophones d'Ingénierie des Connaissances: IC'2002, Rouen, May 2002. Title in english : OntoSpec: a method for the semi-informal specification of ontologies.

[16] KASSEL G., ABEL M.-H., BARRY C., BOULITREAU P., IRASTORZA C., PERPETTE S. (2000). Construction et exploitation d'une ontologie pour la gestion des connaissances d'une équipe de recherche. In *Actes des Journées Francophones d'Ingénierie des Connaissances: IC'2000*, Toulouse, p. 251-259. *Title in english: Design and implementation of an ontology for managing knowledge of a research team.*

[17] KASSEL G., BARRY C., ABEL M.-H. (2000). Programmer au niveau connaissance en Def-*. In J. CHARLET, G. KASSEL, D. BOURIGAULT Eds., *Ingénierie des Connaissances, évolutions récentes et nouveaux défis*, Eyrolle, p. 145-160. *Title in english: Programming at the knowledge level in Def-*.*

[18] KASSEL G., PERPETTE S. (1999). Co-operative ontology construction needs to carefully articulate terms, notions and objects. In *Proceedings of the International Workshop on ontological Engineering on the Global Information Infrastructure*, Dagsthul Castle (Germany), p. 57-70.

[19] MOTTA E., BUCKINGHAM SHUM S., DOMINGUE J. (1999). Ontology-Driven Document Enrichment: Principles and Case Studies. In *Proceedings of the Twefth Workshop on Knowledge Acquisition, Modeling and Management: KAW-99*, Banff (Canada).

[20] O'LEARY D.E. (1998). Using AI in Knowledge Management: Knowledge Bases and Ontologies. *IEEE Intelligent Systems*, May/June 1998, p. 34-39.

[21] RANWEZ S., CRAMPES M. (1999). Conceptual Documents and Hypertext Documents are two different Forms of Virtual Document. In *Proceedings of the Workshop on Virtual Documents, Hypertext Functionality and the Web at the 8th International World Wide Web Conference*, May 1999, Toronto (Canada), p. 21-27.

[22] ROUSSEY C., CALABRETTO S., PINON J.-M. (2001). SyDoM: A multilingual Information Retrieval System for Digital Libraries. In

Proceedings of the 5th International ICCC/IFIP Conference on Electronic Publishing: ELPUB'2001, Canterbury (UK), p. 150-160.
[23] SCHREIBER G., AKKERMANS H., ANJEWIERDEN A., DE HOOG R., SHADBOLT N., VAN DE VELDE W. & WIELINGA B. Knowledge Engineering and Management; The CommonKADS Methodology. MIT Press, 1999.
[24] STAAB S., MAEDCHE A. (2001). Knowledge Portals, Ontologies at Work. *AI Magazine*, Summer 2001, p. 63-75.
[25] USCHOLD M., KING M., MORALEE S., ZORGIOS Y. (1998). The Enterprise Ontology. *Knowledge Engineering Review*, vol 13(1).

An Intellectual Genealogy Graph
~ Affording a Fine Prospect of Organizational Learning ~

Mitsuru Ikeda[1] and **Yusuke Hayashi**[1] and **Hiroyuki Tsumoto**[1] and **Riichiro Mizoguchi**[1]

Abstract. The word of "learning", in a wide sense, is used as a part of the social system of education and it has been attracting researchers' interest in our research area of educational systems. The goal of this research is to support creation and inheritance of organizational intellect, that is, "learning" in an organization. In this paper, we will propose an "Intellectual Genealogy Graph," which is a model representing chronological correlation among persons, activities, and intellect in an organization. The intellectual genealogy graph is a basis of intelligent functions which is useful for surveying current learning conditions and clarifying the intellectual role of individuals, organizations, and documents in the organization.

1. INTORODUCTION

We continue to learn during our lifetimes. As researchers, for example, we learn basic knowledge through 'book learning', acquire up-to-date knowledge from the literature, develop original knowledge for ourselves, and then disseminate it to society. In this sense, we can share the idea that "life is a continuous process of learning." Usage of the word "learning" here has a rather wide sense; it is subtly different from the customary sense in which we use it to refer to the learning process established as a part of the social system of education.

"Learning" in a wide sense includes various forms of learning: for example, workplace learning, life-long learning, organizational learning, and so on. Viewing learning as an implicit, daily, long-term, practical activity is an important trend in many research areas related to the area of computers in education. As examples, the concepts of social constructionism in psychology[1], organizational learning[2] or knowledge creating companies[3] in management, and knowledge management systems in information technology[4][5] have been closely related to our research areas. In our area of intelligent educational systems, needless to say, "learning" in a wide sense has been attracting researchers' interest. Fisher's series of works on life-long learning[6] and integration of collaborative learning and knowledge management[7][8][9] are typical approaches in the same vein.

Along a similar line of thought, this research aims to develop a model of learning in a wide sense. Needless to say, we are all vaguely conscious of a similar model in our own minds which we apply to increase awareness of social relations among organization members; however, that model is implicit and not systemic in most cases. We propose a model called a "dual loop model", which shows how intellect is formed in individual life in organizations and works as a fundamental component of a learning support platform. The dual loop model indicates an ideal relation between individual activity and organizational activity and clarifies roles of individuals, activities, and documents as a vehicle for intellectual communication in organizational learning.

In this research project, we have been developing an IT platform, Kfarm[10][11], to develop users' pro-found social intellectual awareness in organization. Kfarm is a Web-browser-like workplace for users to carry out knowledge-oriented group activities, that is, searching, creating, organizing, and communicating information. All activities on Kfarm are recorded in organizational memory in the form of an "intellectual genealogy graph." This intellectual genealogy graph represents a trace of intellectual activities based on a dual loop model and shows how knowledge and the intellect are evolved in organization.

2. A MODEL OF ORGANIZATIONAL LEARNING

The terms 'knowledge,' 'intellect,' and so on are used with various meanings, so there appear to be no definite meanings for them[12]. Though it is difficult to define them strictly in a consistent manner, to show subjects of this study, we will take some exemplary definitions from the literature.

Brown and Duguid[13] argue convincingly that knowledge is more than just information because it

- usually entails a 'knower',
- appears harder to detach than information, and
- is something what we digest rather than merely hold.

Tobin draws distinctions between data, information, knowledge, and wisdom[14].

1. *Data*:
2. *Information*: = *Data*+ relevance + purpose
3. *Knowledge*: = *Information*+application
4. *Wisdom*: = *Knowledge*+intuition + experience

In this research, the term 'intellect' is used to express our idea similar to Brown and Duguid's argument about 'knowledge' and Tobin's 'wisdom'. Having an intellect means not only merely knowing something, but also digesting it through creation or practical use. It also means that the intellect cannot be separated from a person because it includes skill and competency. Therefore, we aim to support creation and inheritance of organizational intellect by managing information concerned with intellect.

[1] The Institute of Scientific and Industrial Research, Osaka University, 8-1, Mihogaoka, Ibaraki, Osaka, 5670047, Japan

2.1. Organizational learning

It is considered that there are two viewpoints to clarify the goal of creation and inheritance of organizational intellect. One is a practical view and the other is an educational one. The practical goal is to produce a novel and significant intellect for an organization. The educational goal is to properly transmit significant intellect from past to future members of an organization and import significant intellect from outside of it. For both viewpoint, it is necessary to clarify what intellect each organization member has and what kind of shared workplace (Nonaka et al. call this "ba"[15]) makes it easy to transmit each intellect.

We attempt to attain such goals through our usual communication. Typical activities are, for example, acquiring, creating, and distributing intellect through the organization. Linking the activities are vehicles, e.g. conversations, books, or documents. By interpreting the activities and the vehicles, we can gain an awareness of others' intellect; those members usually do various activities to achieve creation and inheritance of organizational intellect based on that awareness. Such individual activities run the organization. However, it is difficult for members to do that because of the implicit nature of an ideal process of creation and inheritance of organizational intellect and content of vehicles actually used in activities. Consequently, to be properly aware of intellect and decide activity to attain the goal, it is necessary to clarify a model representing relations among an organization, individuals, intellect, vehicles, and activities from the view of creation and inheritance of organizational intellect.

Landes et al.[16] proposed a model of organizational learning in which knowledge is augmented with experiences of its application and developed a support tool based on it. The augmentation process is represented by the dependency among the documented experiences. In the best applicable domain of their idea, general knowledge is treated on an abstract level and the essential details of how to apply that knowledge in very specific situations are absent. Basing improvement initiative on experiences has a number of advantages, particularly in such a domain. On the other hand, intellectual roles of a person and intellectual communication in an organization are relatively less focused in this model.

Nonaka and Takeuchi proposed the SECI model, representing a knowledge conversion process and "Middle up-down management", which is a form of an organization to activate process[3][15]. In Middle up-down management, a "Knowledge practitioner (K-practitioner)" plays the role of generating creative power previously mentioned, while a "Knowledge producer (K-producer)" plays the role of coordinating between the top's visions and the K-practitioners' practical activities. Typical activities of the K-producer are given below:

? Proper understanding of organizational conditions.
? Assimilating new intellect with the organizational intellect.
? Distributing organizational intellect based on their vision/strategy.

These activities give direction to K-practitioners' activities.

Several studies have been made on information systems to support creation and inheritance of organizational intellect. Klamma and Schlaphof[17] stated the importance of interrelation between the processes of knowledge creation and usage and normal business processes both on a conceptual and a systemic level; they proposed a model-based approach for solving that. Watanabe and Kojiri[8][18] arranged various kinds of educational support systems: CAI, CAL, ITS, and CSCL according to the

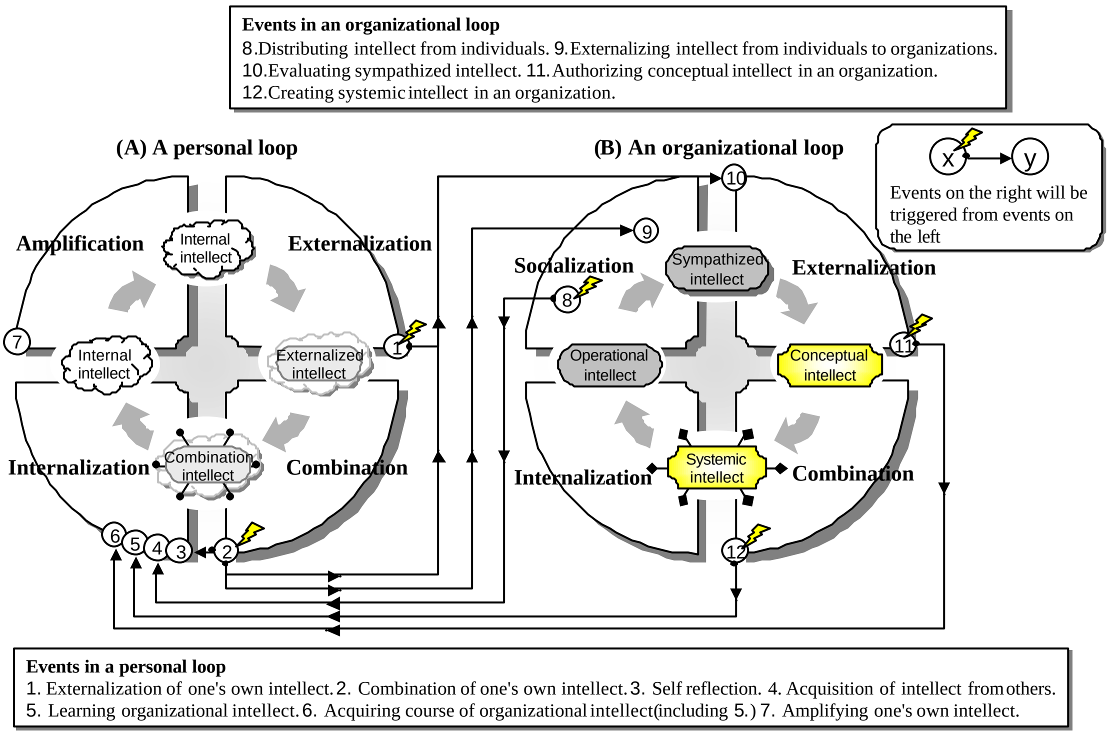


Figure 1. Dual loop model (partly simplified)

SECI model and proposed a learning environment architecture in which learners are able to change their learning style freely. The former study addressed the practical viewpoint and the latter study addressed the educational viewpoint, but each study ignored the other viewpoint.

In the viewpoint of awareness of intellect, Ogata et al. defined awareness of one's own or another's knowledge as "Knowledge awareness" and developed Sherlock II which supports group formation for collaborative learning based on learners' initiatives with the knowledge awareness[19]. This study supports group formation by learners' own initiatives, but lacks the organizational perspective.

The purpose of this study is building an organizational learning environment from several perspectives: practical and educational; and organizational and individual.

2.2. Modeling an organizational intellect

We produced a model supporting creation and inheritance of organizational intellect from two separate models: a process model and a content model. The process model is a model representing creation and inheritance processes of intellect. The content model is a model of the domain of intellectual activities.

Process model. We modeled an ideal abstract process of creation and inheritance of organizational intellect as a "dual loop model". Figure 1 shows the most abstract level of the model, which describes constraint on the relation between activities and change of the property of intellect. For example, socialization prescribes that resultant intellect draws a certain amount of sympathy in the organization; then, externalization of the intellect should follow. These activities are structured as a multi-tiered abstraction hierarchy in which the bottom layer consists of observable activities, for example, reading a document or distributing one. The hierarchy does not prescribe content of intellect concerned with activity, but the property of intellect. The dual loop model explains these activities from both viewpoints of the 'individual' as the substantial actor in an organization (a personal loop: Figure 1(A)) and the 'organization' as the aggregation of individuals (an organizational loop: Figure 1(B)). This model as a whole represents an ideal interrelationship among an organization, its members, and vehicles of intellect for the goal of creation and inheritance of organizational intellect. Further details of the dual loop model are shown in [10].

Content model. Most document management systems manage a document with indexes. However, it is difficult to share it in the organization since the meaning of the indexes is implicit and does not ensure consistency. Even if the document is shared, that will often be done on an implicit premise. In order to share and inherit intellect properly in an organization, it is necessary to form a basis to clarify the meaning of intellect. Semantic web[20] is an attempt to build a global consensus to share resources on the WWW.

Ontology[21] has been brought to public attention as a foundation. Ontology is a set of definitions of concepts and relationships to be modeled. Concepts related to tasks and domains of an organization are defined as the ontology to describe document content. The description is called the "conceptual index". Thus, intellect content in an organization is modeled with an index described on the basis of an ontology.

3. INTELLECTUAL GENEALOGY GRAPH

We compose a model of an organizational intellect as a combination of process and content, that is to say, the dual loop model and the ontology. The model is called an "intellectual genealogy graph". It represents chronological correlation among persons, activities, and intellect in an organization as an interpretation of activities of organization members based on these two models. Modeling an intellectual genealogy graph affords a good foundation for building intelligent support functions for the organizational activities given below.

? Clarifying a role for each member from a trail of his/her intellectual activities in organization. We call the role an "intellectual role", which characterizes a contribution of a person to the construction process of organizational intellect.
? Choosing a correct way to fill a gap between the current condition of organizational intellect and a desired one.

3.1. Components of an intellectual genealogy graph

Principal concepts appearing in an intellectual genealogy graph are as follows:

? Person is a career of intellect and a creator of it.
? Intellect is knowledge, skill, competency, and so on turned to practical use by a person. Categories of intellect are shown in Table 1.
? Vehicle is a representation of intellect and mediates intellect among people. As mentioned before, we assume that intellect can only exist in a person's mind and a vehicle of the intellect is not necessarily a complete representation of the intellect.

Table 1. Types of intellect

Intellect type	Explanation
Personal intellect	An intellect, which a person has personally.
Organizational intellect	Types of intellect classified in view of relation to other's one and organizational one
Sympathized intellect	An intellect consented or sympathized by others
Conceptual intellect	An intellect acknowledged to be significant in an organization
Systemic intellect	A conceptual intellect combined with other conceptual ones.

Table 2. Types of activities (partial)

Activity type		Explanation
Concrete Activity		Observable activities in workplace.
`Read`		Reading, seeing a medium/vehicle.
`Collect`		Collecting a vehicle from other people.
`Represent`		Producing a vehicle.
`Sort`		Sorting a vehicle according to its meaning.
`Distribute`		Distributing a vehicle to other people.
Cognitive Activity		Activities affect on intellect
Personal Activity		Activities concerned with interpersonal activities
	`Create`	Creating new intellect by oneself.
	`Acquire-1`	Acquiring an intellect from others.
	`Organize`	Assimilate a new intellect into his/her own structure of intellect.
Social Activity		An interaction activity as an aggregation of personal activities.
	`Pass`	A person acquires an intellect imparted by another person.
	`Acquire-2`	A person acquires an intellect from on his/her initiative.
	`Discuss`	More than two persons communicate with each other.
Organizational Activity		Activities interpreted in an organizational perspective
	`Share`	Members of the organization share a personal intellect.
	`Authorize`	The organization authorizes a personal intellect.
	`Inherit`	Members of the organization inherit an intellect.

Table 3. Types of relations between intellects(partial)

Relation type	Explanation
created(?a)	A person originally creates an intellect *?a* with no reference to other intellects in the organization.
imported(?a)	A person acquires an intellect *?a* from the outside.
derived(?a, ?b)	A person acquires an intellect *?a* from another person's intellect *?b* in the same meaning.
inspired(?a, ?b)	A kind of *modified* relation, which represents the authorized significance of the conceptual leap from *?a* to *?b*.
authorized(?a, ?b)	A significance of an intellect *?b* is authorized as an organizational intellect *?a* by the organization

? Activity is activity related to the intellect or a vehicle. Categories of activities are shown partly in Table 2.

An intellectual genealogy graph is built by abstracting a causal structure of cognitive activities from concrete activities based on the dual loop model. The structure clarifies mutual relation among personal activities, social activities, and organizational activities.

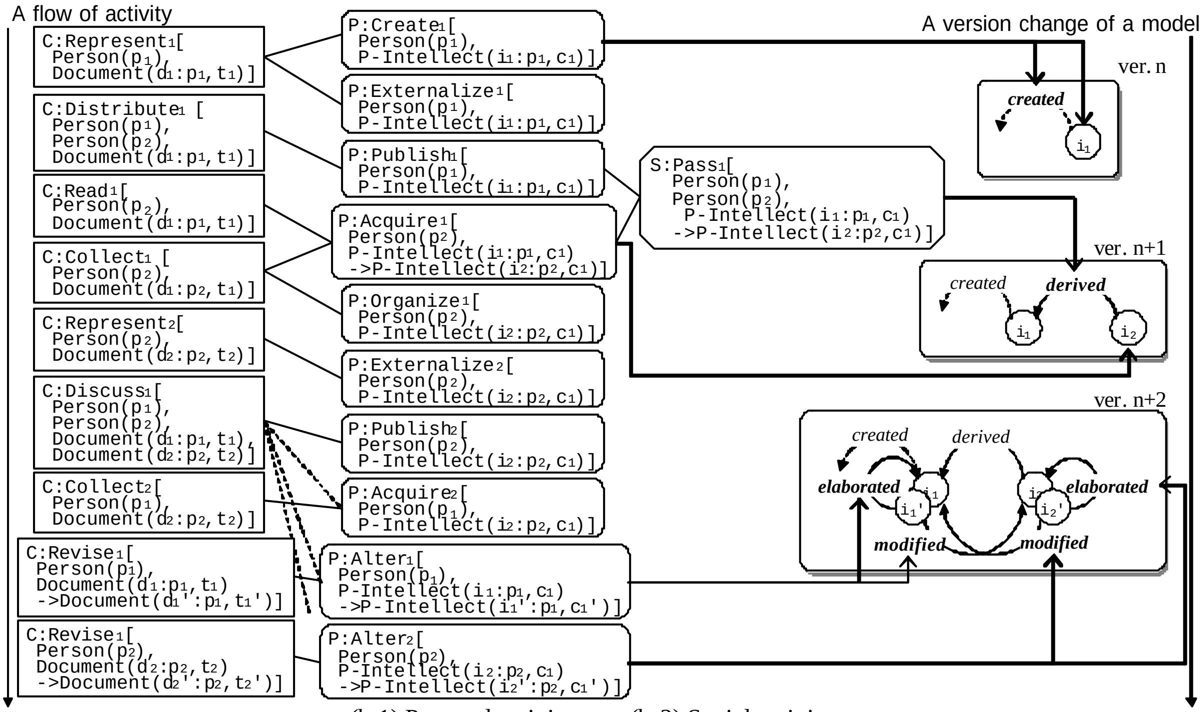
3.2. Modeling an intellectual genealogy graph

An intellectual genealogy graph consists of a vehicle layer and an intellect layer. The vehicle layer comprises persons, vehicles, and concrete activities. On the other hand, the intellect layer is an interpretation of the vehicle layer and consists of persons, intellects, cognitive activities, and relations among intellects. These relations are classified into some types by characteristics of changes of intellect as shown in Table 3. In the intellect layer, these relations are built from activities.

Hard data for modeling an intellectual genealogy graph is a time-series of concrete activities observed in the workplace. Firstly, a vehicle layer of the graph is built from the data. Then, a series of cognitive activities are abstracted from the vehicle layer based on the dual loop model and an intellect layer of the graph is constructed. Figure 2 shows an example of interpretation from concrete activities into cognitive activity and relationships between intellects derived by the translation. In this way, the intellectual genealogy graph records the formation of an organizational memory from activities.

4. Kfarm: AFFORDING FINE PROSPECT OF INTELLECTUAL ACTIVITIES

Kfarm is a system that we have been developing which embodies our conceptualization thus far. Kfarm is a distributed system consisting of a K-granary, at least one K-ranch house and some K-fields. The K-field and the K-ranch house are environments for a K-practitioner and a K-producer respectively. Those two play dual roles of sensors which watch a user's activities in a knowledge-oriented task and a display which shows information



(b-1) Personal activity (b-2) Social activity

(a) Concrete activity (b) Cognitive activity (c) A model of intellect

Figure 2. An example of an intellect genealogy graph

about the organizational intellect according to their roles. The K-granary is a server. It interprets K-producers' and K-practitioners' activities observed in the K-field and the K-ranch house and then aggregates and stores them as an organizational intellect.

4.1. K-field

A K-field provides K-practitioners with information needed for their knowledge-intensive tasks. Typical K-field functions are given below. These are designed based on activities defined in the personal loop in the dual loop model.

Sorting documents by folders: A K-field provides a bookmark window as a tool to store documents in folders with indexes. The indexes are converted to conceptual indexes in the K-granary.

Communication with others: In a KW-window, a K-field indicates information about others and documents related to the document selected in the bookmark window. This information is based on intellectual roles of members and the document assigned on the intellectual genealogy graph.

4.2. K-ranch house

A K-ranch house supports K-producers' activities, e.g., recognizing the organizational condition and coordinating communication, cooperative work, and collaborative learning between K-practitioners based on the organizational vision/strategy.

Figure 4 shows windows of the K-ranch house which is under development. A launcher window shown in Figure 4(A) informs K-producers about activities of K-practitioners in Kfarm. Figure 4(B) and (C) are monitor windows to provide a K-producer with detailed information of an organizational memory. In this case, an icon shown in Figure 4(A-1) indicates growth of an intellect supposed to be a sympathized intellect. If the K-producer clicks this icon, its details will be shown in the monitor window as shown in Figure 4(B) and (C). Figure 4(B) graphically indicates who sympathizes with the intellect through which document. Each node in Figure 4(C) indicates an intellect. Links between them indicate relations between intellects previously mentioned in Table 3.

Now, we will take a close look at the visualized intellectual genealogy graph. Figure 4(C) indicates a history of a generation of intellect in which the intellect (C-1) is the center of attraction. Broken arrows from intellect (C-2) to (C-1), for example, indicate an elaborated link. It is interpreted from the fact that ikeda makes a document referring to hayashi's document concerned with intellect (C-2) and puts the same term index and additional ones on the document. This information help the K-producer to clarify intellectual roles of members and documents concerned with the intellect. To illustrate a case of this, for example, it is supposed that hayashi is a person who has made a seed of a new intellect (C-2) and documents concerned with intellects (C-3) can be used as background information.

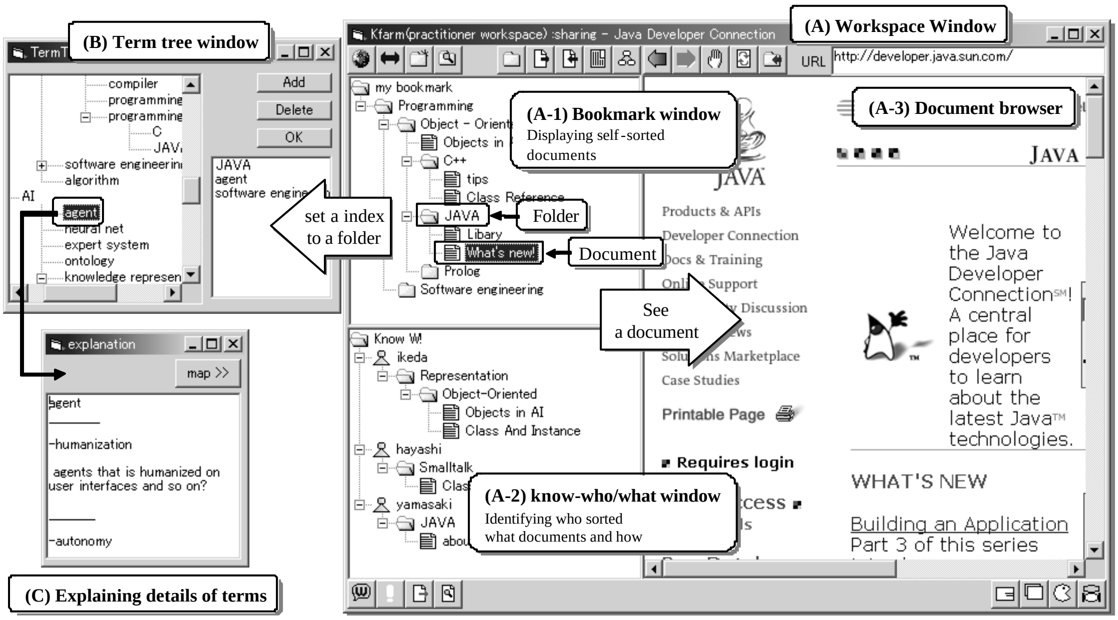


Figure 3. K-field

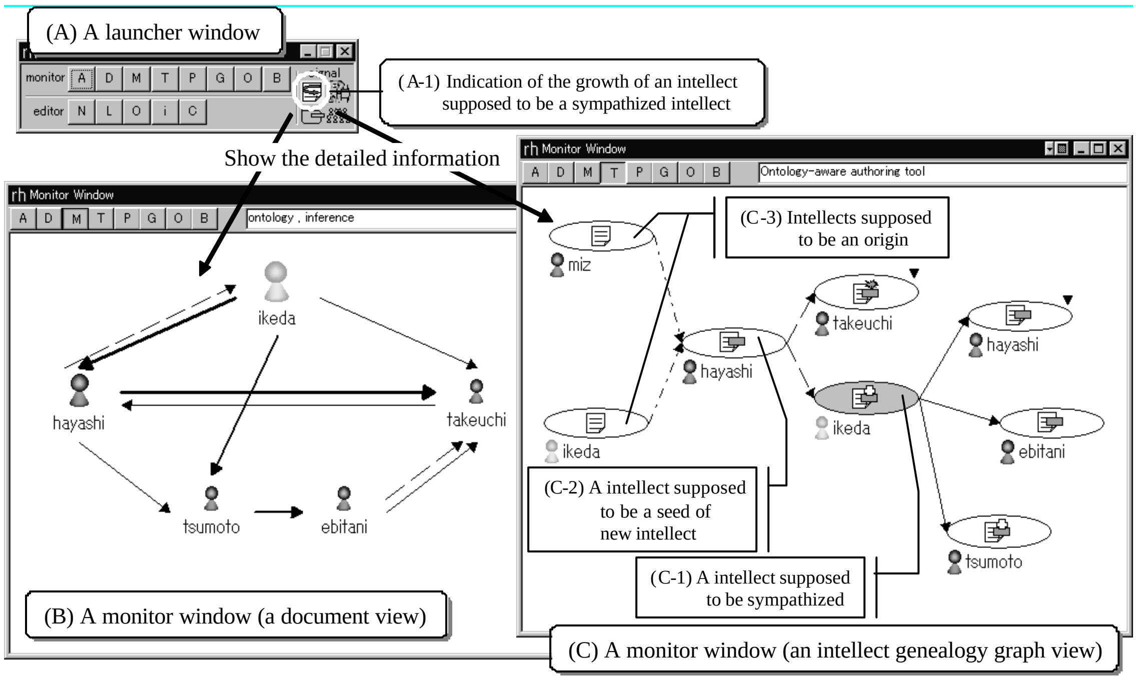


Figure 4. K-ranch house (under development)

5. CONCLLUSION

In order to support creation and inheritance of organizational intellect, that is, "learning" in a wide sense, it is important to abstract and interpret activities in the organization. In this paper, we have proposed the dual loop model and ontology as bases and introduced Kfarm as an embodiment of them. The intellectual genealogy graph is useful for individuals and organizations to survey current learning conditions and to clarify the intellectual role of individuals, organizations, and documents in the organization.

Future direction of this study will be to augment Kfarm in the following two ways.

? Support of arranging a collaborative learning space

? Model of the property of an organization

In the former, broadly speaking, it is considered that Kfarm itself is a space for less-regulated collaborative learning because it allows learner-directed communication. However, some processes of a dual loop model can be better achieved by rather regulated collaborative learning.

In the latter, generally, an organization has a hierarchical structure and a member belongs to some groups in the structure. Currently, we are introducing an organizational structure and developing a more flexible model of creation/inheritance of organizational intellect by considering that structure.

REFERENCES

[1] Doise W. and Mugny G.: The social development of the intellect, Oxford: Pergamon Press, 1984.

[2] Senge P. M.: The Fifth Discipline: The Art and Practice of Learning Organization, Doubleday, 1990.

[3] Nonaka I. and Takeuchi H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, 1995.

[4] Decker S. and Maurer F.: "Editorial: Organizational memory and knowledge management", Int. J. Human-Computer Studies, 51, pp. 511-516, 1999.

[5] Fensel D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag Telos, 2001.

[6] Fischer G.: "Lifelong Learning - More Than Training, Special Issue on Intelligent Systems/Tools in Training and Life-Long Learning". In Eds.: Mizoguchi R. and Piet A.M. Kommers, Journal of Interactive Learning Research, Vol. 11, No. 3/4, pp.265-294, 2000.

[7] Eleuterio M. A., Bortolozzi F., and Kaestner C.A.: "The Roles of Ontologies in Collaborative Virtual Learning Environments", Proc. of ECAI2000 Workshop on Analysis and Modeling of Collaborative Learning Interactions, pp.31-35, 2000.

[8] Watanabe, T.: "Knowledge Management architecture of Integrated Educational Support", Proc. of ICCE/SchoolNet 2001, pp.1138-1141, 2001.

[9] Ayala G.: "Intelligent Agents Supporting the Social Construction of Knowledge in a Lifelong Learning Environment", Proc. of NTCL2000, pp.79-88, 2000.

[10] Hayashi Y., Tsumoto H., Ikeda M., and Mizoguchi R.: "Toward an Ontology-aware Support for Learning-Oriented Knowledge Management", Proc. of ICCE/SchoolNet 2001, pp.1149-1152, 2001.

[11] Hayashi Y., Tsumoto H., Ikeda M., and Mizoguchi R.: "Kfarm: A Knowledge Management Support System Based on Dual Loop Model", Proc. of PYIWIT'2002, pp.235-242, 2002.

[12] Liebowitz J. (Eds.): Knowledge Management Hand Book, CRC Press, 1999.

[13] Brown J. S. and Duguid P.: "The Social Life of Information", Harvard Business School Press, Boston, 2000.

[14] Tobin D.: Transformational Learning: Renewing Your Company thought Knowledge and Skills, John Wiley and Sons, 1996.

[15] Nonaka I., Toyama R., and Konno N.: "SECI, Ba, Leadership: a Unified Model of Dynamic Knowledge Creation", Long Range Planning, 33, pp. 5-34, 2000.

[16] Landes D., Schneider K., and Houdek F.: "Organizational learning and experience documentation industrial software projects", Int. J. of Human-Computer Studies, 51, pp.643-661, 1999.

[17] Klamma R. and Schlaphof A.: "Rapid Knowledge Deployment in an Organizational-Memory-Based Workflow Environment", Proc. of ECIS 2000, pp. 364-371, 2000.

[18] Kojiri T., and Watanabe T.: "HARMONY: Web-based Adaptive Collaborative Learning Environment", Proc. of ICCE/SchoolNet 2001, pp.559-566, 2001.

[19] Ogata H., Matsuura K., and Yano Y.: "Active Knowledge Awareness Map: Visualizing Learners Activities in a web Based CSCL Environment", Proc. of NTCL2000, pp.89-97, 2000.

[20] W3C Semantic Web: http://www.w3.org/2001/sw/

[21] Mizoguchi R. and Bourdeau J.: Using Ontological Engineering to Overcome AI-ED Problems, Int. J. of Artificial Intelligence in Education, Vol.11, No.2, pp.107-121, 2000

Initiating Organizational Memories using Ontology Network Analysis

Yannis Kalfoglou[1], **Harith Alani**[1], **Kieron O'Hara**[1], **Nigel Shadbolt**[1]

Abstract. One of the important problems in organizational memories is their initial set-up. It is difficult to choose the right information to include in an organizational memory, and the right information is also a prerequisite for maximizing the uptake and relevance of the memory content. To tackle this problem, most developers adopt heavy-weight solutions and rely on a faithful continuous interaction with users to create and improve its content. In this paper, we explore the use of an automatic, light-weight solution, drawn from the underlying ingredients of an organizational memory: ontologies. We have developed an ontology-based network analysis method which we applied to tackle the problem of identifying communities of practice in an organization. We use ontology-based network analysis as a means to provide content automatically for the initial set-up of an organizational memory.

1 Introduction

Organizational memories (hereafter, OMs), have been studied as means for providing easy access and retrieval of relevant information to users. There are several technologies which support the implementation and deployment of OMs (some of them identified in [1]), however, there is relatively little support for the initial set-up of an OM. When implementing and deploying an OM, it is difficult to identify the right information to include. This task is, normally, a knowledge engineer's job, to identify relevant information and populate the OM accordingly. This process though, is time-consuming, manual and error-prone given the diversity and quantity of resources to be analyzed for relevance. Semi-automatic methods and techniques exist, but these are bound to individual technologies, as for example in [1] where the authors state that: "the knowledge engineer [then] integrates the information obtained from the thesaurus generator into the OM semi-automatically, scanning the similarity thesaurus and deciding which relations should be formalized and added to the knowledge base or ontology, which should be included in the thesaurus integrated with the ontology, and which should be ignored". On the other hand, it is always the user who has to "kick off" search in the OM. This however, requires the user to formulate a query, sometimes with the help of semi-automatic support, and then the OM system has to parse the query successfully, retrieve information deemed to be relevant according to some pre-defined notion of relevance, and present it to the user.

Another perception on OMs is in terms of knowledge delivery. There have been two, metaphorically-defined, ways of delivering knowledge reported in the literature: 'pull' and 'push' knowledge [35]. The former refers to technologies which aim at pulling knowledge from vast repositories of data to people. Examples include the familiar search engines which, in some implementations, are facilitated by intelligent agents augmented with ontologies for semantically-enriched search (see, for example, the OntoSeek [21] and FindUR [30] systems). In these systems the user is expected to initiate the search by posing queries. On the other hand, 'push' systems aim at providing knowledge to their users without prior interaction. Means to achieve this ambitious goal in knowledge management (hereafter, KM) is the focus of semantically-described content, the identification of the user's task and task context.

In OM applications, both ways have been studied, though the 'pull' technologies seem to be dominant. The reason for the low uptake of 'push' technologies in knowledge delivery is probably the increased risk of 'bombarding' the user with irrelevant information which in turn could result in dissatisfaction with and discrediting the OM. To tackle this problem, OMs that used 'push' technologies made certain assumptions. For example, the KnowMore OM [2] assumes that an existing workflow engine will be in place; this in turn will be accessed and linked to the OM making it possible to reveal context-specific information regarding the user's task. Having such information available before initiating search, could (semi-)automate the task of filling-in queries with context-specific information. That way, knowledge deemed relevant to the process is proactively presented to its user.

Although we found this marriage of workflow processes and OMs an interesting one, we are skeptical about two, often unforeseen, obstacles in deploying such a system: (a) there might be situations where processes will not be easy to identify or codify in a workflow engine and (b) even when these are available and the OM is built around existing processes, it might not be desirable to restrict a user's search on those resources that are deemed to be relevant to the process the user is involved in. In addition, the technological challenges OM developers face when implementing this merger of workflow processes and OMs could be considerable [3].

To alleviate this situation, we are exploring the use of one of the core technologies for supporting OMs, that of ontologies. In particular, to cope with the problem of initially setting up an OM, we apply a method used in the Advanced Knowledge Technologies (AKT) project, Ontology Network Analysis (hereafter, ONA). We apply an algorithm to identify objects that are more important than others in the underlying ontology. We measure importance in terms of popularity. Those that have been identified are used as the initial seed to populate the OM, thus setting-up an OM containing some information readily available for use. Since our method is based on an ontology, we take advantage of the underlying ontological structures to draw inferences on the objects selected and reason about the rel-

[1] Advanced Knowledge Technologies (AKT), Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK, email: {y.kalfoglou,kmo,ha,nrs}@ecs.soton.ac.uk

evance of retrieved information. We applied this method to tackle an OM problem: how to identify communities of practice (hereafter, CoP).

This automation in initially setting up an OM does not eliminate the user from the picture. We are keen to explore the synergy between user-defined input and automatically-delivered content. To achieve this, we worked on ways to customize the ONA, allowing the user to customize the output of the automated content-delivery mechanism. We explored these issues in the context of our testbed application, CoP.

We give an overview of the work related to initiating OMs by emphasizing reported trade-offs between user-defined queries and (semi-)automatic query definition in section 2. We then continue with an objective analysis of the resources selection problem when setting-up an OM (section 3) which motivates our hypothesis in section 4. We test our hypothesis in section 5 with a comprehensive case study applying ONA to harvest information about a valuable OM resource: CoPs. We generalize the approach in section 6 and we discuss further implications of this approach to supporting OMs in 7 where we also point to future work.

2 Related work

In the KnowMore OM [2], means for semi-automatically constructing the underlying ontologies were investigated. The authors describe an interactive thesaurus-based methodology for ontology construction which is realized in a designated editor. Their focus is on extracting (semi-)automatically an ontology from domain-specific texts. In addition, the characterization of knowledge items to be used in the OM is supported by automatic tools which attach meta-data to the text. This will be used in later phases of an OM's lifecycle for guiding the retrieval and storage of related information. In our ONA-based approach we are not focusing on how to construct the underlying ontologies. As we will describe in section 4, we assume these have been constructed beforehand. Our focus is on how to provide as much information as possible to the OM user for initial set-up. However, there is an overlap of interests and methods with the (semi-)automatic ontology construction work done in AKT reported in [44].

The work described in [27] is the closest to the ONA approach. The authors describe the information retrieval process as a "select" operation on database query languages with appropriate search conditions formulated with respect to "(i) meta-data given in the information ontology (which information resources to consider or how old information to retrieve), (ii) specific-context information (employ sophisticated similarity measures for comparison of actual query situation and context factors of knowledge sources described in the OM), and (iii) the content searched for.". However, their retrieval techniques are based on annotations and their similarity measure algorithms explore only one dimension of the underlying ontology network: the subsumption dependence between nodes, i.e., class-subclass relationships. To allow users to customize their search, they provide application-specific heuristic search based on the notion of 'heuristic expression'. The users can formulate their own heuristic search formulae based on a standard template formula which takes as input a set of nodes of the underlying directed graph and for each node follows the links specified in the formula in a left to right order, delivering at each step an intermediary set of nodes as a new starting point for the next step. Although this option allows users to customize their search, the actual retrieval is based on the same subsumption mechanism. On the other hand, as we describe in section 5, ONA allows a multidimensional traversal of nodes in the ontology network with thresholds, traversal paths, and starting nodes being user-defined, if desired.

In [6], Atlhoff and colleagues propose a method for OM Improvement (OMI). They argue for a method which supports user feedback as a way of improving OM over time. In their comprehensive analysis of factors that determine the usefulness of an OM they identified the selection of knowledge to be included in the OM as an important one:

> "[conceptual knowledge] determines what and how experience stored in the OM plays a major role regarding the usefulness of a system."

They continue by arguing that "users often do not bother with too many questions, a problem which usually arises during the initial set-up of the OM". The conceptual knowledge Althoff and colleagues are referring to is the underlying ontology in our ONA-based approach. To tackle the problem of initial set-up, we use ONA to populate the OM automatically with the most important objects as identified from their popularity in the underlying ontology. As in [6], we also intend to use a characterization of the object to be displayed in the OM along with its popularity value as obtained from the ONA. This textual information is much appreciated by OM users [6], as it gives them explanations of the selected information. Since we base our method on an ontology, we could easily obtain these characterizations from standard 'documentation slots' which exist in most ontology development environments.

Cohen and colleagues [12], were among the first to investigate the use of metrics for ontologies. In the context of the HPKB US project [14], ontology metrics were defined to measure the level of reuse of ontological concepts in applications. For example, whenever a new axiom was added in the application's knowledge base, the metric calculated the ratio of reuse of existing ontological concepts in the newly added axiom. For ONA we use a spreading activation algorithm to all ontology constructs and do not define specific metrics.

3 The problem of resources selection

Despite the research reported above, a major problem when initially setting-up an OM remains unsolved: how to select the right resources to include in an OM? This problem has been identified in field surveys [17] as well as in implemented systems (e.g.: [2], [6]). This is a multi-faceted problem because it is not only concerned with the elicitation of resources that will be presented to the user or used for retrieving relevant information. These resources are also often:

- used by other systems within the organization, which incidentally also serve users in their quest for valuable information;
- 'unspecified', in that they are vaguely expressed, need to be composed by a number of related resources or are external to the organization;
- and once these resources are identified and put into use they act as a qualitative measure for the OM.

That is, if an OM's users are not satisfied with the quality of information presented to them, it is unlikely that they will return, especially when there are other conventional information-seeking systems in the organization that users used to use before confronted with an OM.

A way of tackling this resource-selection problem is by identifying the purpose of the OM: what are the users' needs and what will the OM be used for. This has been reported as one of the first phases in building an OM [17]. The techniques and methods for achieving this rather ambitious goal are mostly taken from requirements analysis and elicitation research. They stem from Computer Supported Collaborative Work (hereafter, CSCW) research, from systems design research, and from the cognitive science literature.

However, we should be cautious when we are calling upon requirements engineering to elicit the needs when building an OM. As Zave and Jackson report in their survey [47], vague and imprecise requirements are always difficult to formalize and subsequently convert to specifications, in the early phases of software development[2]. This refinement is necessary, the authors continue, "to bridge the gap between requirements and specifications", thus emerging with a specification that could satisfy users' needs and meet the requirements. In the case of OMs, we should expect these requirements to be incomplete and vague. In addition, as Dieng and colleagues report in [17], building OMs presumes that we will re-use methods, approaches and techniques we have applied in the past in other domains:

> "(1) corporate memories are not entirely new systems; they are adaptations, evolutions or integrations of existing systems; (2) before conceiving memories, the proponents or users of the solutions have taken part in the design of other types of systems (knowledge-based systems, CSCW systems, etc.), and they have transferred the solutions they already know. Most of the solutions can thus be considered as adaptations of existing solutions."

The vagueness and incompleteness of requirements from prospective OM users led some designers to decide to build their OM around an existing workflow process engine, as for example in the KnowMore OM. We discuss the adaptability of this approach and its advantages of achieving a 'near perfect' integration with existing IT organizational infrastructure and satisfying users' (pre-defined) needs further in section 7, but for now we would like to focus on the importance of having a 'comprehensive' OM from its initial set-up. By comprehensive we mean an OM that includes a lot of resources that have been automatically extracted rather than waiting the user to initiate the extraction process. The side-effect of having this sort of OM in place is that we can tackle the 'cold start' syndrome identified in [19] in which the authors reported that they had relatively few knowledge assets in their OM during the first operational month which led to low access rates from its users as they couldn't see the value-added of the OM. The problem was eventually solved, but at a cost: more systems and methods had to be used to chase users for contributions in order to enrich the content of the OM, thus leading to an increase in the OM's knowledge assets and consequently in increased access figures.

In the following section we elaborate how our method sets up a comprehensive OM in an automated fashion.

4 Seeding the OM

The basis of our solution is ontologies. These consensual representations of the important concepts in some domain of interest have been studied, developed and deployed for over a decade now in various fields and applications in academia and industry. Their use in OMs has been advocated in field surveys [1] and in applied OMs (see, for example the KnowMore OM [2], the EULE2 system [40], or the integration of ontologies and Experience Factories, a form of OM, for improving maintenance [23]). Our hypothesis is that since we already use ontologies in OMs for the purposes of semantic interoperability and reuse, we could also use them in other ways. We could analyse their structure by taking into account relationships between their constructs, based on a tunable spread activation algorithm, yielding the nodes that are most "popular". These are assumed, in the absence of contradicting evidence, to be the most important ones. The spreading activation algorithm also identifies nodes similar to a specific node. This is the premise underlying our hypothesis.

It could be argued that our analysis is not a qualitative one, but merely a quantitative one. However, as Cooper argues in [16], quality can be measured in two ways, in terms of popularity or importance. Our analysis yields concepts that are the most popular in the network, and since the network is about an ontology which by default represents important concepts, then these concepts are also important.

To operationalize our hypothesis, we assume that (a) ontologies will be available in the organization in which we want to deploy an OM, and (b) these will be populated. It is clear that these assumptions are strong and indeed are ongoing research issues in the knowledge engineering community, especially the latter. However, we should accept and anticipate that ontologies are popular in organizational settings nowadays, in the form of database systems, other knowledge sharing formalisms more common to the AI research community (e.g.: KIF) or indeed in emerging semantic web standard formats (e.g.: RDF(S)). As an open research issue, we are already in AKT investigating ways of (semi-)automatically constructing ontologies.

Using ontologies as the foundation for an OM is not a unique idea, but the use of ONA to provide initial information for populating the OM is novel. We should also mention that using an ontology at the start of an OM's lifecycle allows us to provide support to users in formulating their queries from an early stage. Normally, users have to formulate initial queries unaided since there is no prior information available, as no retrievals have been made yet. In applying ONA, we support users in formulating queries by providing them with ontological information regarding the starting node for initiating an ONA-based search. This information is readily available in existing slots in the underlying ontology (such as the documentation slot).

5 ONA

In this section, we set out the principles underlying ONA, and then demonstrate an application of the method — gathering information on CoPs. In section 5.2, we then set out the opportunities and problems that characterize the study of CoPs. Finally, in section 5.3, we set out an application of ONA to the problem of kick-starting an OM for a particular CoP.

5.1 Principles of Ontology Network Analysis

ONA [5] is the technique of applying information network analysis methods to a populated ontology to uncover certain trends and object characteristics, such as shortest paths, object clusters, semantic similarity, object importance or popularity, etc. A variety of such methods have been explored in the past for different information retrieval purposes. ONA investigates the application of these methods to analyse the network of instances and relationships in a knowledge base, guided by the domain ontology. There are many methods of studying networks, and of course many types of networks that can be studied (cf. [33]). However, the advantage of studying ontologies is that the relations therein have semantics or types, and therefore that the semantics provide another source of information over and above connectivity or simple subsumption. This semantic information can be taken account of when performing a network analysis, allowing

[2] In our case, the early phase of developing an OM.

"raw" results to be refined on a relatively principled basis. An ONA example application is described in section 5.3 and an example algorithm is detailed in [5].

ONA methods can be harnessed to address the resources selection problem in building OMs (section 3), by using populated ontologies already in place in organizations to select a set of important and interesting resources to feature in a new OM. The fact that the method is automatic takes some of the burden of OM development from its users or managers, and allows some quality content to be put in place prior to use, thereby increasing the likelihood of early take-up by its users.

Being automatic, ONA is not, of course, foolproof or infallible. Many points of interest in an organization's ontology will not be spotted by the methods involved, especially if the ontology is in some way incomplete, and fails to cover the object domain fully in some important respect. Clearly, ONA cannot be the *only* principle used to populate an OM. However, by extracting some information from an ontology, ONA can be used to suggest an initial set of interesting concepts and relations. Certain assumptions must be made to support the use of ONA here, but as the OM develops, such assumptions can be relaxed, as the population of the OM begins to happen by its users. And user feedback as to the actual importance of the entities uncovered will always be essential.

The ONA technique of interest to this paper is the application of network measures to an ontology to determine popular entities in the domain. Such entities can be either classes or instances, where *popularity* is (a) defined in terms of the number of instances particular classes have (class popularity), and the number and type of relation paths between an entity and other entities (instance popularity), and (b) regarded as a proxy for importance. Clearly this latter claim is one that will not always be true. However, the working assumption is that the important objects will have a stronger presence in a representation of the domain, and will have a lot of key relationships with many other entities (they will act as "hubs" in the domain)[3].

Given a first pass ONA of an ontology, giving the most popular entities, an OM developer can exploit user feedback to hone the analysis. Two particular ways of doing this can be envisaged.

1. Important instances can be selected — these instances may have been counted as 'popular' under the first pass analysis or not, as the case may be, and hence could be manually selected as important instances independently of the governing assumption that popularity = importance — and the ONA performed once more, this time measuring not the quantity of relations between all entities, but measuring the quantity of relations between the selected instances and other entities.
2. Relations can be weighted according to their importance, and the weights transferred from entity to entity along the relation-connection. Hence one relation (e.g. *co-author-with*) might be weighted more highly than another more common one (e.g. *shares-office-with*), whose relevance to the domain in question is not as high. In that case, the effect when performing an ONA is to privilege the entities that enter into the highly-weighted relations as against those that do not. There are two (classes of) ways of differentially weighting relations.
   (a) First, relations could be differentially weighted automatically, on similar lines to the selection of important entities, viz., the relations most often filled with values in the knowledge base will be weighted higher than others.
   (b) Alternatively, the weights can be fixed manually. This has the advantage of being sensitive to user understanding of the domain, and the disadvantage of being a complex and difficult process that could be time-consuming, especially if there are a lot of relations about. Of course, as with entity-selection, an initial cut using automatically-created weights could be run past a user, who might suggest adjustments; this might be the cheapest method of getting the best of both worlds.

In the next subsection, we discuss communities of practice, and then we go on to examine the use of a particular spreading activation algorithm to perform an ONA in order to extract information about communities that is latent in a domain ontology.

5.2 Communities of Practice

CoPs' value in the construction and maintenance of OMs has been acknowledged by other OMs developers. To quote [4]:

> "[. . .] employees solve knowledge-intensive tasks (KITs) cooperatively as a community of practice, embedded into the overall business workflows and supported and monitored by the OM system. Applications used and repositories filled and queried are the closely related basis of the OM environment, and value-added care about intelligent support for knowledge indexing, distribution, storage, search, retrieval, and integration."

A CoP is an informal group of individuals with a common interest in a particular work practice. Their interest should take a particular form: the individuals concerned should wish to *improve* their practice, either for financial reasons (picking up bonuses, or securing promotion), or mere professional pride. The CoP then plays a number of roles. First, the individuals in it will meet informally to discuss particular problems and issues facing the practice; in this way the CoP fosters a common appreciation and characterisation of the practice. Second, particular solutions will be demonstrated and evaluated within the CoP; the CoP therefore fosters innovation, partly through the shared understanding of problems, and partly through the evaluation "process," which is likely to be rigorous and competitive. Third, the informal nature of the contacts mean that, almost automatically, innovations will be built on by interested others who "tinker with" or improve them; informality means that restrictive practices such as patenting or licensing tend not to be invoked within the CoP, and therefore that innovation very naturally becomes a collaborative process. Fourth, new exponents of the practice can use the CoP as an important tool for situated learning of the practice. After training, most effective learning takes place "on the job," as new practitioners discuss their problems with their fellows, or learn from their colleagues how to integrate the practice with the rest of their business workflow; in such a way, the CoP becomes a repository and dissemination mechanism combined for best practice [45].

A CoP contrasts with other, more formal structures that centre round a practice [15].

- *Functional groups* specialize in particular functions within an organization, for example, marketing, administration, security or finance. The agents form a homogeneous set, drawn together by

[3] One doubtless common circumstance where this assumption will *not* be reliable would be where an ontology is pieced together from legacy datasets. In such a case, the most popular entities are likely to be those represented in detail elsewhere for other purposes, whose importance may not carry over into the current application. Another point to note is that quantitative information may be more prevalent than qualitative information, and that therefore entities that enter into many quantitative relations could be overvalued. We emphasise once more: user feedback is essential.

disciplinary specialization, and are organized in hierarchies; the purpose of the group is not to produce learning, though of course new recruits achieve situated learning. The hierarchical structure, and often a shared educational background, keeps the group together.

- *Teams* are also well-defined within organizations. They are made up of individuals brought together to carry out a given task, each chosen because of some specialist skill that is assumed to be required for the task's performance. Hence the members of a team are highly heterogeneous, and the team's management will be intended to integrate their functional knowledge. Learning, if it takes place, is unintended, and tends to be via the interactions across functional specialities — a specialist might come to understand the constraints on, and the requirements and responsibilities of, his colleagues. The team's life is normally not extended beyond the achievement of the task's goals.
- A *network* consists of individuals across organizations who have interests in working together, for example, in a rough system of producer interests, some of whom provide components, parts or expertise for a final manufacturer; the function of the network is to bring together suppliers and consumers of particular goods or services to facilitate negotiations, or to cut purchase overheads, e.g. information-gathering costs. Such a network is made up of heterogeneous agents, and focuses on the exchange of knowledge, perhaps encoded in price signals. The requirement for complementary knowledge keeps the community going, and a necessary condition of this is a high level of mutual trust.
- *Epistemic communities* are relatively formal groups of agents who produce knowledge, or codes for expressing knowledge, from some position of authority that may be formal (e.g. a professional association), or more informal (e.g. based on particular agents' positions of eminence). Different interests tend to insist on representation in such forums, and hence the makeup of such a community can be quite heterogeneous. Such communities often play a wider political role, and can be the "public face" of a discipline. Recruitment to such groups is founded on peer approval.

In contrast to these types of group, CoP's members — it has an informal, self-selecting, largely homogeneous membership — are interested in increasing their skills, and in accumulating and circulating best practice. As a result, a CoP is an excellent vehicle for situated learning of the practice [45].

When we consider which types of group are of interest for OMs, then the comparison is very instructive. Organizational learning has a dual aspect [8]. "Single-loop learning" is an organizational learning process whereby knowledge is obtained to solve problems based on an existing and well-understood model of the domain, in other words a routine process. "Double-loop learning" involves the establishment of a new set of paradigms, models, premises, representations or strategies to supersede the existing models, to improve the organization's response to existing problems, and to enable the organization to address new problems. These types of learning are called "Learning I" and "Learning II" by Bateson [9].

As Nonaka and Takeuchi point out ([31], p.45), one problem with the adoption of this approach to learning — useful as it has been in a number of respects — is that it sees organizational learning as a process of adaptation to external stimuli that involves the development and modification of existing routines supported by OM, not as a process where knowledge is created. Even when such a view is taken, it can be difficult for insiders to spot the right moment for attempting serious knowledge creation, except by making such a process routine — in which case of course there is no guarantee that there will be no period when either (a) knowledge acquirable only by double-loop learning is required but not available, or (b) an expensive double-loop learning process is initiated for which there is no immediate requirement.

Part of the trouble is that much learning theory, as in epistemology generally, has as its focus the individual [10, 32]. The problem here is that when this focus is transferred to actual cases of organizational learning, the complexity of the collective learning process, which cannot straightforwardly be reduced to a simple addition of learning processes for the individuals in the organization, cannot be properly respected. The key to implementing effective organizational learning processes is to understand the organization in terms of the collectives that make it up, the overlapping groups that were listed above; learning across these organizations, then, is a complex process of interaction between these heterogeneous entities [10, 45, 15].

One important role for OM, therefore, is to act as the information storage buffer between these overlapping groups. In that event, a key factor from the point of view of creating or seeding an OM is the availability of various resources. *In general*, the more formal a group, the more likely it is that relatively tractable sources are available for populating an OM. There are two reasons for this: first, formal functions lend themselves to careful management that can track events and leave a highly visible audit trail, and second, their very formality places those events on the management radar. In contrast, informal groups, by their nature, are often undetected by management, and their "memory" may well boil down to the sum of the non-metaphorical psychological memories of their members, with all the potential problems that this implies.

In particular, a functional group, say, or a team, is barely likely to have a life outside of their working existences. For example, the former has a strict hierarchical structure, which regulates the permitted interactions between members — a division of labour intended to increase efficiency — to a series of delegations, as the task is understood at increasingly lower levels of abstraction as we move down the hierarchy. Each level of the hierarchy might well, by contrast, form a CoP, and may have links not only across equivalent nodes in the functional group hierarchy, but also with equivalent levels in hierarchies of orthogonal functional groups within the organization, or with similar levels in related functional groups in other organizations. Here the CoP, parasitic on the functional group, is formed by people wishing to understand the process of, in this case, receiving a task description at one level of abstraction, and decomposing it into subtasks which can then be delegated to available resources further down. The OM of the functional group will consist of the decompositions and delegations, together with the feedback that passes up the hierarchy; creating and maintaining such an OM is, of course, non-trivial. But the OM of the CoP is not something that will spontaneously appear, consisting as it does of informal chats and retellings of "war stories" around the photocopier or in the pub after work.

Similar considerations apply to teams and epistemic communities. Each consists of heterogeneous agents brought together to carry out a particular task, or open-ended series of tasks in the case of the epistemic community. In that event, the actual work of the team or epistemic community generally takes place in formal scheduled minuted meetings. Converting this relatively stable resource to an OM proper is, no doubt, problematic in various ways, but there is at least a fairly straightforward way to begin to populate the OM. On the other hand, the informal work done that pertains to the team goes on within related CoPs. Team members go back to their informal CoPs, transmitting new knowledge about the requirements of people who carry out different functions, and tinkering with new ways to incorporate such exogenous requirements. The knowledge created by a team or epistemic community is analogous to the gears of an engine, whereas the knowledge of the CoP is analogous to the oil; the former is much more visible than the latter, but will eventually seize up and grind to a halt if the latter is not present.

As a result of such considerations, CoPs are seen as key elements in the efficient working of an organization, and as key agents in knowledge management [45, 18, 34]. Well-known companies that have nurtured CoPs include Hewlett-Packard Consulting, Arthur Andersen, Accenture, Ernst and Young, BP, Caltex, Chevron, Conoco, Marathon, Mobil, PDVSA, Shell, Statoil, TOTALFINAELF, Intel, Lucent, Siemens, Xerox, IBM, the World Bank and British Telecom [41, 26]. Smith and Farquhar give a detailed example of the use of CoPs in the oil industry consultants Schlumberger [41]. Schlumberger supports the development and maintenance of an OM for its oil engineering CoP by providing what is called a knowledge hub, consisting of a series of technologies designed to support worldwide connectivity between those engineers, and to foster a culture that encourages its use; such technologies are relatively straightforward — email, the web, bulletin boards, together with data management systems, project archives, expertise directories and so on. Maintenance of the different parts of the knowledge hub is detailed specifically to *knowledge champions*, people responsible for animating the community, encouraging participation, reporting successes etc. ([41], pp.22–27). Smith and Farquhar are clear about the importance of populating such resources.

> "Just because an intranet portal has been built filled with world-class technology, it is not a given that community members will flock to it. Do not overwhelm them with all the features that computer scientists can think of that "clearly" would be beneficial. Instead, be cautious. Determine first what technology the community members actually use. . . .
> An up-front investment is required to seed the initial knowledge repository. It is difficult, if not impossible, to convince community members to contribute to an empty shell. . . . Not only must there be content from the launch date, but it must be quality content as well." ([41], p.28)

This vision of the creation of a CoP memory beginning with a seeding process is shared by Marshall and colleagues [28], where their concept of a *community memory*, the open-ended set of knowledge and shared understandings that acts as the CoP's intellectual glue, maps pretty well onto the CoP OMs that we have been discussing. The daily activities of the CoP members are seen refracted through this community memory. The problem, as they see it, is that as the community develops, the memory grows so that the maintenance task becomes overwhelming; simultaneously, however, the memory is growing stale, with inconsistencies, redundancies and irrelevancies proliferating as the focus of the CoP changes, and as the CoP needs to maintain contact with exogenous sources of knowledge, such as the web or other large-scale information resources. In that case, there will have to be a process of purging, together with a restructuring of a trimmed down OM.

However, such seeding restructuring processes, as advocated by [41, 28], are rendered much more complex by the informal nature of the CoP itself. Too firm a smack of management will destroy the informal nature of the CoP — and therefore make it much more difficult for the CoP to support the invisible, informal parts of the work process [45]. CoP management is a delicate process, and various methods have been suggested for doing it [46, 29]. These methods all begin with one of the most difficult aspects of managing informal communities — discovering the extent of the community itself.

5.3 ONTOCOPI

To this end, we have applied a particular instantiation of ONA to attempt to isolate CoPs within organizations described by ontologies [33]. The rough idea is to use an ontology-based spreading activation algorithm to search the knowledge base, moving from instance to instance along relationship connections as defined by the ontology. The system is called ONTOCOPI (ONTOlogy-based Community Of Practice Identifier), and is currently implemented as a Protege ([20]) plug-in as well as a standalone Web accessible program.

Spreading activation was first introduced by Quillian [38] to simulate human semantic processing in a machine subsequently it has formed the basis for many information retrieval methods such as semantic similarity measures, Web analysis algorithms, community identification, case-based reasoning, etc. ONTOCOPI's algorithm combines and improves ideas from previous work on similarity measures, such as shortest path measures [39], multi-path traversal [36], and constrained spreading activation methods [13]. ONTOCOPI's algorithm can make use of the ontology to make decisions about which relationships to select and how they should be valued. Ontological axioms can also be consulted in the relationship selection process.

Some caveats must be pointed out here. Relationships in ontologies are mostly of a formal nature. CoPs however, tend to have an informal nature, which is one of the major difficulties for CoP management (section 5.2). The traditional method used to identify CoPs most often[4] appears to be more or less structured interviewing ([46], pp.8–10) and recently Sol and Serra proposed a multiagent Web-based approach ([42]). The ONTOCOPI assumptions about CoP identification attempt to get around this time consuming activity.

A formal relationship can stand as proxy to an informal one. Hence we can infer that two people who co-author a paper are more likely to be members of the same CoP. If two CoP members actually share no formal relationships (at least, no formal relations captured by the ontology), then any vector addition of formal relations can also stand proxy for informal ones. Hence if A co-authored a paper with B, who works on a project with C, then it may be inferred that A and C, who have no formal connection, are more likely to be members of the same CoP. Total accuracy, of course, is impossible for an informal and rapidly-evolving social group like a CoP; furthermore, the aim of ONTOCOPI is only to *support* CoP identification, a very expensive operation in its own right [46]. A certain measure of indeterminacy is inevitable.

Another fact of importance is that ONTOCOPI can't identify relationships that aren't there: if two people in the same CoP simply have no formal relationship recorded in the ontology, and no chain of formal relations linking them, then their co-membership cannot be found. The information has to be in the ontology for ONA to tease it out. Finally, ONTOCOPI can't distinguish *between* CoPs. If someone is a *broker*, i.e. a person who functions in two separate CoPs [45], then ONTOCOPI will tend to pick up the union of the two CoPs (although the settings can be modified somewhat to try to ameliorate this difficulty — see below).

It follows that ONTOCOPI cannot infallibly identify a CoP. But then a CoP is in many ways indeterminate anyway. ONTOCOPI, however, does support CoP identification, a resource-heavy task that

[4] Except in organizations *defined* around a CoP, which may include Schlumberger [41].

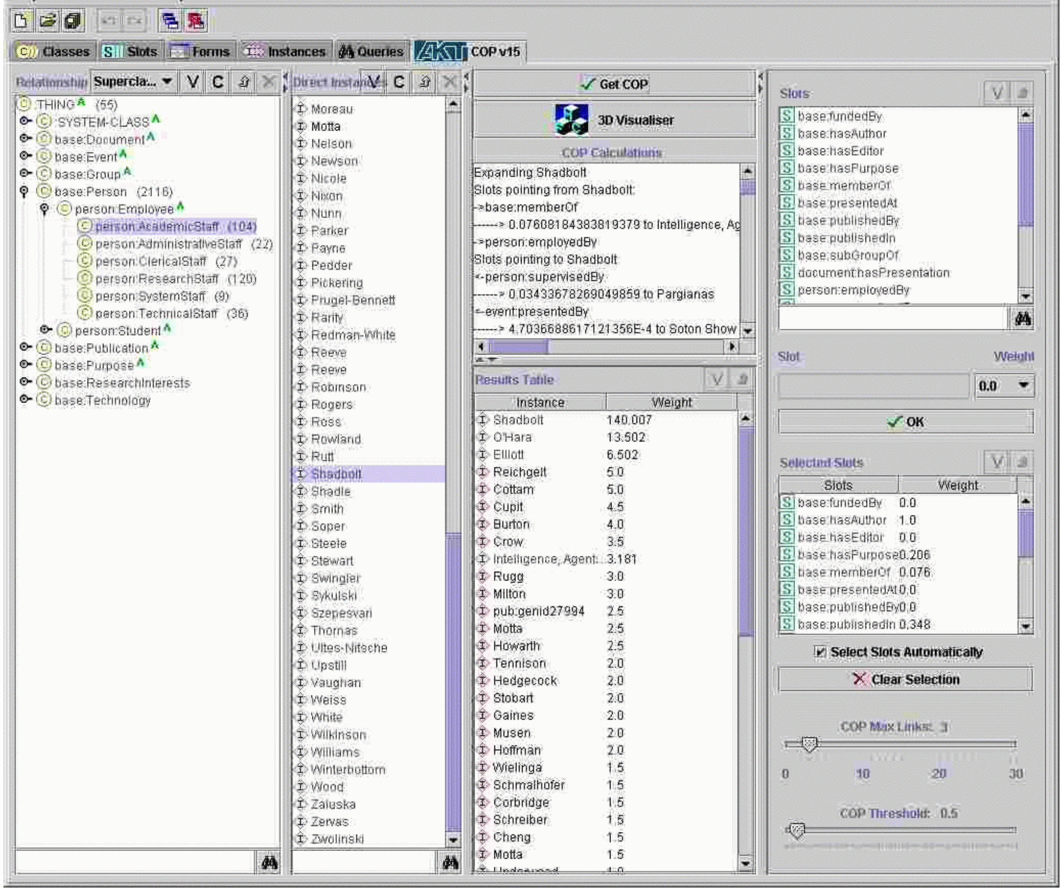


Figure 1. A screenshot of ONTOCOPI as a Protege plug-in.

may be alleviated to some extent by the not-so-subtle assumption that formal connections can approximate informal relationships.

The interface can be seen in figure 1. As a prototype, we do not claim that this is in any way optimal, but it indicates the information it can give. The panel on the far left shows the class hierarchy of the ontology. The panel next to it shows the instances of a selected class. From this panel, an instance can be selected to be the "centre" of the CoP investigation (i.e., the relations radiating out from this individual will be those used as the basis of the CoP identification). The panels on the right hand side set the relation weights and parameter values (e.g., the number of links the algorithm will spread to). Clicking the 'Get COP' button will set the algorithm going. The centre right top panel displays the current calculations, and centre right bottom displays the weights that have been transferred to other instances, in descending order of weight (i.e. a rough specification of the CoP, the main output of ONTOCOPI). In this diagram, the CoP of *Shadbolt* has been investigated, and ONTOCOPI has suggested, in descending order of preference, *O'Hara*, *Elliott*, *Reichgelt*, *Cottam*, *Cupit*, *Burton* and *Crow*, then the *Intelligence, Agents, Multimedia Group* of which *Shadbolt* is a member, then *Rugg* and so on.

Order is important, so are the relative weights. *O'Hara* scores 13.5; this is meaningless except in the context of a particular search. Here, 13.5 is very good, twice the score of the next candidate. On the other hand, the user may be more suspicious of the ordering of, say, *Tennison*, who scores 2.0, and *Motta*, who scores 1.5. The figures themselves have no constant interpretation (except in terms of the algorithm); it is for the users to take the suggestions and interpret them according to their own understanding of the structure of their CoP. Hence ONTOCOPI, to reiterate, only supports CoP identification.

The relation weights can be created automatically based on frequency, or created artificially. In this run, the weights were calculated automatically, with the most frequently used relation getting weight 1, those not used at all getting 0, and the others being allocated accordingly. This, then, might be a first run; a second run might adjust the weights manually, perhaps giving some less used but important relations higher weights.

The algorithm initializes instance weights to 1, and then applies a breadth-first spreading activation search, going through all the relations, and using the relation weight and the instance weight of the departure node, transfers more weight to the arrival node. It then continues the search, this time out from the arrival node. Instances then accumulate weight according to the numbers of relations (or chains of relations) they have with the initial instance chosen to start the process; the longer the chain, the smaller the weight transferred; the weightier the relation, the larger the weight transferred. Hence a short distance, or a significant connection, with the base instance will tend to push an instance up the batting order. In the example, *O'Hara* has written a lot of papers with *Shadbolt* — many individual relations of a highly significant kind in this context (indeed this paper by its very existence has already increased *O'Hara's* score, as well as those of *Alani* and *Kalfoglou*). *Shadbolt* has few direct connections with *Gaines*, but their transitive links are many and varied, and hence

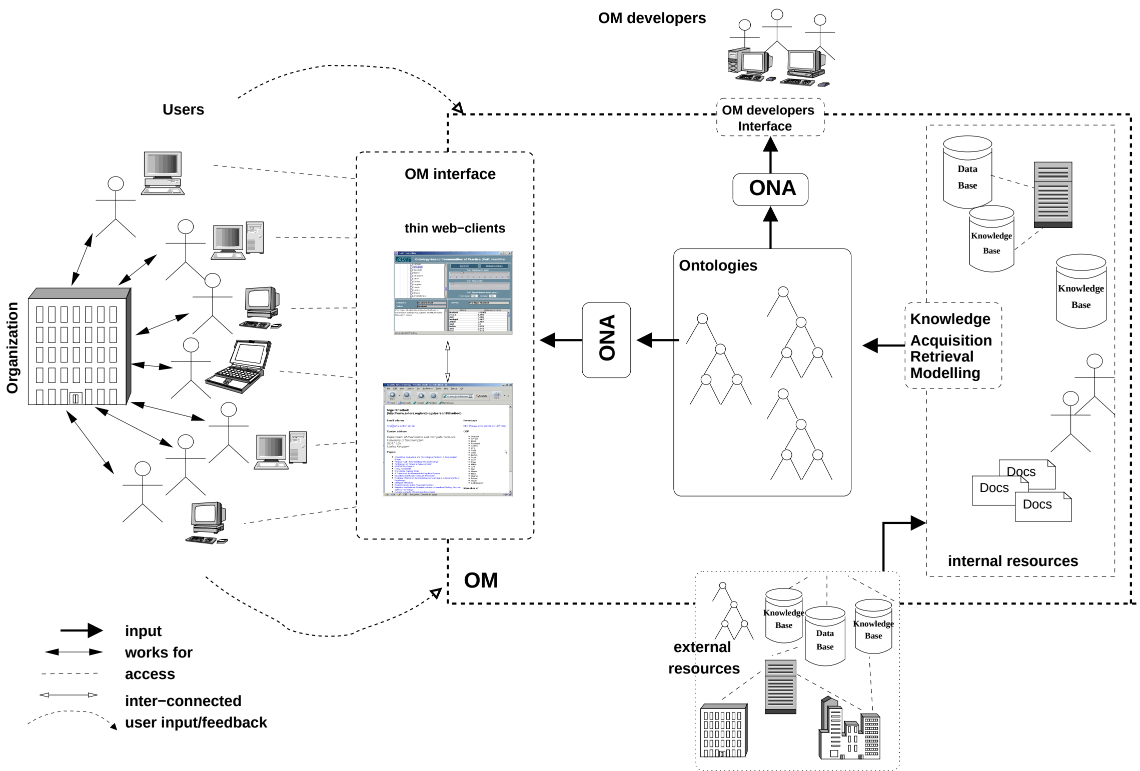


Figure 2. Applying ONA at different phases of OMs: to push knowledge to users as well as help developers tune their OMs.

Gaines appears on the radar.

The "raw" algorithm can be refined according to user feedback — recall that user feedback is essential with ONA. Manual setting of relation weights has already been mentioned. Other ways to control variables include:

- *Temporal considerations*, if they are modelled in the ontology, can be factored in. For example, the relations might only be considered if they were extant, say, in the last 5 years. [5] shows how, on this interpretation, *Shadbolt's* CoP has altered over the last fifteen years, beginning in the mid 80s with a number of psychologists, who gradually fall out of the picture as we move towards the present, when AI and later knowledge engineering and KM concerns take over as *Shadbolt's* academic career evolved; new people become colleagues, or become connected to *Shadbolt* by other more or less circuitous routes.
- *Filtering out "hubs"*. One problem, already implicitly mentioned, is that of "hubs". A hub, in this context, is a highly-connected person with lots of relations with other people through work, publishing, or whatever. Such people carry a lot of relative weight — in more ways than one — and so can sometimes skew the CoP by transferring an inordinate amount of weight to the instances with which they are connected. The ONTOCOPI algorithm can constrain the weight transfer based on the level of connectivity of such people. This allows the comparison of CoPs to see what contribution certain people made to them.
- *Privileging of classes*. Particular classes can be selected to identify the concepts of interest, and then the system will automatically select the relationships that interconnect these classes, and assigns relationship weights on the basis of their frequency.
- *Differential initial weighting of instances*. This is not implemented yet, but one could imagine altering the initial weights, either manually (selecting definite CoP members and ruling out definite non-members, and increasing the value/devaluing all their relationships accordingly), or automatically (e.g., increasing the weights of papers which contained certain key words in their titles or abstracts).

One could imagine many more adjustments to refine the basic picture. The appropriate refinements in a particular domain will depend on the features of the domain itself, and what is captured by the ontology.

We have described one way to apply ONA to the problems of resource selection for OMs. In the next section, we move on to a generic account of the relation between ONA and OMs.

6 Generalising the method

In figure 2 we depict a high-level diagram of an OM. This is not meant to be a reference architecture for OMs, such as the one depicted in [25]. This figure emphasizes the dual role of ONA and the supportive role ontologies play in our scenario. On the left-hand side of the figure we have users of an organization performing their regular tasks. In the centre we have an OM which is composed, at this abstract level, by two interfaces to users and OM developers, a port to external resources, and internal resources existing in the organization's repositories. The latter could have several forms, ranging from tacit knowledge possessed by experts to explicit knowledge expressed formally in KBs or databases. In the centre of our abstract OM, lie the ontologies which underpin the entire OM. These are either existing resources or are constructed (semi-)automatically with the aid of knowledge acquisition, retrieval and modelling techniques. We do not refer to these in this paper as our focus is on the use of ONA: the two rectangular boxes denoting "ONA" are placed between the ontologies and OM interfaces to users and developers. The genericity of ONA makes it possible to use it for pushing knowledge to users but also as an aid for the OM's developers. They could apply ONA to the organization's ontologies in order to identify which concepts should be presented to certain types of users. For instance, assuming that there is a workflow engine in the organization, and developers are looking for ways of linking the OM to it, they could either engage in modelling techniques such as those used in linking the KnowMore OM with workflow processes [2], or they could use ONA to help them identify which concepts from the underlying ontologies are mapped onto the ones of the workflow's processes. This activity requires inspection and familiarization only with one end of the prospective link: that of the workflow processes. The developer then, uses the concepts found in the workflow processes as a starting node for his/her ONA. This could reveal whether further linking is feasible (or otherwise), thus saving development time and allowing developers to deal with ontologies that they are not familiar with. The approach taken by the KnowMore OM, requires a careful analysis and possibly, modelling of workflow processes and ontologies before a link between them could be implemented. ONA can ease the analysis on the ontology end of this prospective link.

We also include two curly dotted arcs in figure 2 linking users with the OM. These denote users' feedback and input. This is an important, probably the most important, element of any OM architecture. As Althoff and colleagues have shown in [7], an OM can be improved over time by user feedback and input. In our abstract architecture, we envisage light-weight feedback mechanisms, implemented as thin Web-clients, accessible through Web browsers, as a means for eliciting feedback on an OM's resources. An example of such technology from the AKT project is the Digital Document Discourse Environment [43] used as a digital discussion space.

Finally, the OM interface to its users is light-weight and accessible from distributed clients on the Web. We have developed several such interfaces for accessing our dedicated tools in AKT. An example, taken from the CoP application (section 5.3) is illustrated in figure 3. Two kinds of interfaces included here: a dedicated OM interface, where the user can state preferences in selecting the appropriate node to search for related information, or there could be a customized rendering of information into a user's Web browser. The latter is extracted automatically after applying ONA to the underlying ontology, whereas the former requires user input to tune the search criteria.

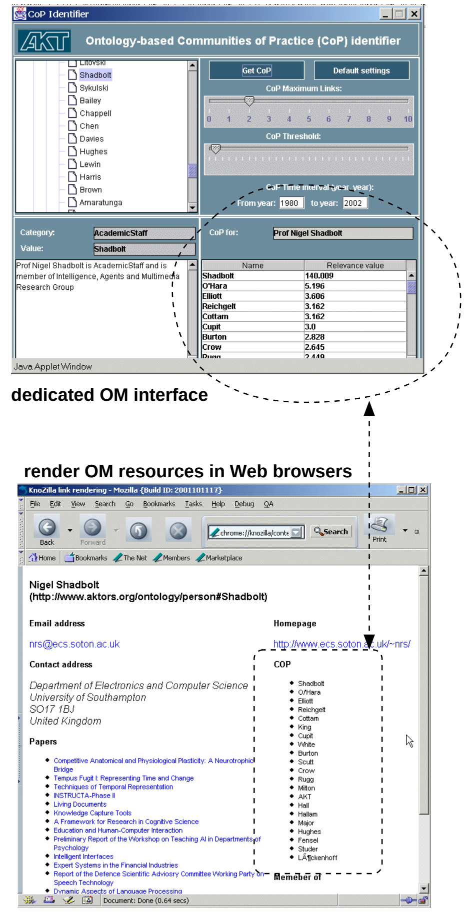


Figure 3. Different ways of accessing OM's resources: through dedicated Web-run interfaces or via standard Web browsers.

7 Discussion and further work

In this section we elaborate on some implications and potential caveats of our ONA. We categorize them in three broadly defined areas: information overload, context-awareness and domain-independence. We critically review the application of ONA when these areas are considered in deploying OMs:

- **Information overload:** As Abecker and colleagues pointed out in their KnowMore OM, the progressive and query-based interaction with the OM from initial set-up acts as "a safeguard against unwanted information overload."[2]. Potential drawbacks include: progressive interaction means that the initial set-up will suffer from 'cold-start' syndrome, not enough information will be available; query-based interaction requires expertise and domain familiarization from the users to get the most out of an OM. The advantages are discussed below under the heading 'context-awareness'. There isn't a golden rule to follow when we, as developers, face this dilemma. It is worth pointing out though that users, amid the bulk of information ONA pushes to them, are still in control of it. They can change the search criteria (namely, the starting node in the ONA algorithm), to meet their preferences. Users can also choose which relations to traverse and their relative importance (weights). Further, we support this change as much as possible by ontologically-guiding the user in choosing the right starting node, as nodes always carry some sort of semantic information drawn automatically from the underlying ontology. So, it could be argued, this task becomes a pedagogical experience for users apart from easing their query formulation.

- **Context-awareness:** this has been recognized as the Achilles' heel for OMs. One proposed remedy, advocated by proponents of marrying workflow processes and OMs (see, for example [3]), seems to work well in settings where workflow processes are either existing, or are relatively easy to identify and model. ONA takes a different approach in tackling context-awareness. We do not assume that workflow processes will exist, but we merely rely on ontological resources which we assume exist or could be constructed. Contextual relevance can be achieved in a number of ways thanks to the genericity of ONA. We could rely on ad-hoc technologies, such as profiling users' interests by using agents [37] or by embedding personalization facilities in thin Web clients [24], or rely on identification of users' tasks [11]. In addition, our reliance on organizational ontologies gives us the ability to exploit knowledge about users identity (obtained from system-entry logs), and thus help guess their information needs.
- **Domain-independence:** this is a desired feature for OMs. ONA is not specific to any kind of ontology, or indeed to any ontology at all! This makes it possible to apply ONA to more than one ontology as are likely to exist in large organizations. As we described in the previous section, we could use ONA as a tool to assist knowledge engineers in deciding which ontologies to consider for supporting the OM. This in turn, speeds-up the task of selecting appropriate organizational ontologies. However, ONA will not be the only tool to be used in this process: in the case of similar or conflicting ontologies there might be a need to integrate them or to resolve inconsistencies. In this case, ONA is only one of the many tools that knowledge engineers would like to have at their disposal to tackle these challenges.

A number of components described in this paper are not fully implemented yet. As this is ongoing work, we are in the process of integrating several tools developed in the context of the AKT project to realize the generic architecture described in section 6. We have already designed, developed and deployed the CoP exemplar application in various settings and are currently in the process of evaluating it. We have also developed much of the infrastructure needed to deploy such an OM: Web clients [24] and ontologies are ready for use. We are currently working on methods for maintaining these ontologies, constructing and populating them as automatically as possible [44]. Several application scenarios are currently under consideration one of which would use OMs to access heterogeneous resources and push information to dedicated members of a community. In these scenarios we plan to use the knowledge-sharing infrastructure developed in AKT [22].

ACKNOWLEDGEMENTS

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. The views and conclusions contained herein are those of the authors and should not be intepreted as necessarily representing official policies or endorsements, either expressed or implied, of the EPSRC or any other member of the AKT IRC.

REFERENCES

[1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, and M. Sintek, 'Toward a Technology for Organizational Memories', *IEEE Intelligent Systems*, **13**(3), 40–48, (June 1998).

[2] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, and M. Sintek, 'Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project', *International Journal on Information Systems Frontiers (ISF), Special Issue on Knowledge Management and Organizational Memory*, **2**(3/4), 139–162, (2000).

[3] A. Abecker, A. Bernardi, S. Ntioudis, G. Mentzas, R. Herterich, C. Houy, S. Muller, and M. Legal, 'The DECOR toolbox for workflow-embedded organizational memory access', in *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS'01), Setubal, Portugal*, (July 2001).

[4] A. Abecker, A. Bernardi, and M. Sintek, 'Proactive knowledge delivery for enterprise knowledge management', in *Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering(SEKE'99), Kaiserslauten, Germany*, pp. 120–127, (June 1999).

[5] H. Alani, K. O'Hara, and N. Shadbolt, 'ONTOCOPI: Methods and tools for identifying communities of practice', in *Proceedings of the 2002 IFIP World Computer Congress, Montreal, Canada*, (August 2002).

[6] K-D. Althoff, A. Birk, S. Hartkopf, W. Muller, M. Nick, D. Surmann, and C. Tautz, 'Managing Software Engineering Experience for Comprehensive Reuse', in *Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslauten, Germany*, pp. 10–19, (June 1999).

[7] K.-D. Althoff, M. Nick, and C. Tautz, 'Improving organizational memories through user feedback', in *Proceedings of the Learning Software Organizations(LSO'99) workshop, Kaiserslauten, Germany*, ed., F. Bomarius, pp. 27–44, (June 1999).

[8] C. Argyris and D. Schon, *Organisational Learning: A Theory of Action*, Addison-Wesley, 1978.

[9] G. Bateson, *Steps to an Ecology of Mind*, Paladin, 1973.

[10] J.S. Brown and P. Duguid, 'Organizational Learning and Communities of Practice: Toward a Unified View of Working, Learning and Innovation', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 99–121. Butterworth-Heinemann, (2000). first published 1991.

[11] J. Budzik and K.J. Hammond, 'User interactions with Everyday Applications as Context for Just-In-Time Information Access', in *Proceedings of the 2000 International Conference on Intelligent User Interfaces, ACM*, (2000).

[12] P. Cohen, V. Chaudhri, A. Pease, and R. Schrag, 'Does prior knowledge facilitate the development of knowledge-based systems?', in *Proceedings of the Sixteenth National Conference on Artificial Intelligence, AAAI'99, Orlando, FL, USA*, pp. 221–226, (July 1999).

[13] P. Cohen and R. Kjeldsen, 'Information retrieval by constraint spreading activation in semantic networks', *Information processing and management*, **23**(4), 255–268, (1987).

[14] P. Cohen, R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Gunning, and M. Burke, 'The DARPA High Performance Knowledge Bases project', *AI Magazine*, **19**(4), 25–49, (1998).

[15] P. Cohendet, F. Creplet, and O. Dupouet, 'Interactions Between Epistemic Communities and Communities of Practice as a Mechanism of Creation and Diffusion of Knowledge', in *Interaction and Market Structure: Essays on Heterogeneity in Economics*, eds., D. Delli Gatti, M. Gallegati, and A. Kirman. Springer-Verlag, (2000).

[16] W.S. Cooper, 'On selecting a measure of retrieval effectiveness', in *Readings in Information Retrieval*, pp. 191–204, (1997).

[17] R. Dieng, O. Corby, A. Giboin, and M. Ribiere, 'Methods and tools for corporate knowledge management', *International Journal of Human-Computer Studies (IJHCS)*, **51**, 567–598, (1999).

[18] N.M. Dixon, *Common Knowledge: How Companies Thrive By Sharing What They Know*, Harvard Business School Press, 2000.

[19] C. Gresse-vonWangenheim, D. Lichtnow, A. vonWangenheim, and E. Comunello, 'Supporting Knowledge Management in University Software R&D Groups', in *Proceedings of the 3rd International Workshop on Learning Software Organizations (LSO'01), Kaiserslautern, Germany*, eds., K-D. Althoff, R. Feldmann, and Muller, volume 2176 of *Lecture Notes in Computer Science*, pp. 51–66. Springer-Verlag, (September 2001).

[20] W. Grosso, H. Eriksson, R. Fergerson, J. Gennari, S. Tu, and M. Musen,

'Knowledge Modelling at the Millennium - The design and evolution of Protege2000', in *proceedings of the 12th Knowledge Acquisition, Modelling, and Management(KAW'99), Banff, Canada*, (October 1999).

[21] N. Guarino, C. Masolo, and G. Vetere, 'OntoSeek: Content-Based Access to the Web', *IEEE Intelligent Systems*, **14**(3), 70–80, (May 1999).

[22] K. Hui and A. Preece, 'An Infrastructure for Knowledge Communication in AKT', White paper v.1.0, available from URL: http://www.csd.abdn.ac.uk/research/akt/aktbus-1.0/index.html, Department of Computer Science, University of Aberdeen, (July 2001).

[23] Y. Kalfoglou, 'On the convergence of core technologies for knowledge management and organisational memories: ontologies and experience factories', in *Proceedings of the ECAI 2000 Workshop on Knowledge Management and Organisational Memories(W11-KMOM'00), Berlin, Germany*, (August 2000).

[24] Y. Kalfoglou, J. Domingue, E. Motta, M. Vargas-Vera, and S. Buckingham-Shum, 'MyPlanet: an ontology-driven Web-based personalised news service', in *Proceedings of the IJCAI'01 workshop on Ontologies and Information Sharing, Seattle, USA*, (August 2001).

[25] O. Kuhn and A. Abecker, 'Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges', *Journal of Universal Computer Science*, **3**(8), 929–954, (1997).

[26] E.L. Lesser, M.A. Fontaine, and J.A. Slusher, 'Preface', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. vii–xi. Butterworth-Heinemann, (2000).

[27] M. Liao, A. Abecker, A. Bernardi, K. Hinkelmann, and M. Sintek, 'Ontologies for Knowledge Retrieval in Organizational Memories', in *Proceedings of the Learning Software Organizations(LSO'99) workshop, Kaiserslauten, Germany*, ed., F. Bomarius, pp. 19–26, (June 1999).

[28] C.C. Marshall, F.M. Shipman III, and R.J. McCall, 'Making Large-Scale Information Resources Serve Communities of Practice', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 225–247. Butterworth-Heinemann, (2000). first published 1995.

[29] R. McDermott, 'Why Information Technology Inspired But Cannot Deliver Knowledge Management', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 21–35. Butterworth-Heinemann, (2000). first published 1999.

[30] L.D. McGuinness, 'Ontological Issues for Knowledge-Enhanced Search', in *Proceedings of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98), Trento, Italy*, ed., N. Guarino, pp. 302–316. IOS Press, (June 1998).

[31] I. Nonaka and H. Takeuchi, *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, 1995.

[32] K. O'Hara, *Plato and the Internet*, Icon books, 2002. ISBN: 1840463465.

[33] K. O'Hara, H. Alani, and N. Shadbolt, 'Identifying Communities of Practice: Analysing Ontologies as Networks to Support Community Recognition', in *Proceedings of the 2002 IFIP World Computer Congress, Montreal, Canada*, (August 2002).

[34] K. O'Hara and N. Shadbolt, 'Issues for an Ontology for Knowledge Valuation', in *Proceedings of the IJCAI'01 workshop on E-Business and the Intelligent Web, Seattle, WA, USA*, (August 2001).

[35] D. O'Leary, 'Using AI in Knowledge Management: Knowledge Bases and Ontologies', *IEEE Intelligent Systems*, **13**(3), 34–39, (June 1998).

[36] C.D. Paice, 'A thesaural model of information retrieval', *Information processing and management*, **27**(5), 433–447, (1991).

[37] Ph. Perez, 'Corporate Memory Management through Agents', in *Proceedings of the eBusiness and eWork Conference, Madrid, Spain*, (2000).

[38] M.R Quillian, 'Word concepts: A theory and simulation of some basic semantic capabilities', *Behavioral Science*, **12**, 410–430, (1967).

[39] R. Rada, H. Mili, E. Bicknell, and M. Blettner, 'Development and application of a metric on semantic nets', *IEEE Transactions on systems, management, and cybernetics*, **19**(1), 17–30, (1989).

[40] Reimer,U. and Margelisch,A. and Novotny,B. and Vetterli,Th., 'EULE2: A knowledge-based system for supporting office work', SIGGROUP Bulletin 19(1), ACM, (1998).

[41] R. Smith and A. Farquhar, 'The Road Ahead for Knowledge Management: An AI perspective', *AI Magazine*, **21**(4), 17–38, (2000).

[42] R. Sol and J. Serra, 'Netexpert: A multiagent system for expertise location', in *Proceedings of the IJCAI-01 Workshop on Knowledge Management and Organizational Memories, Seattle, USA*, (August 2001).

[43] T. Summer and S. Buckingham-Shum, 'From Documents to Discourse: Shifting Conceptions of Scholarly Publishing', in *proceedings of the CHI'98: Human Factors in Computing Systems, Los Angeles, CA, USA*, pp. 95–102. ACM Press, (1998).

[44] M. Vargas-Vera, J. Domingue, Y. Kalfoglou, E. Motta, and S. Buckingham-Shum, 'Template-driven information extraction for populating ontologies', in *Proceedings of the IJCAI'01 Workshop on Ontology Learning, Seattle, WA, USA*, (August 2001).

[45] E. Wenger, *Communities of Practice: The Key to Knowledge Strategy*, Cambridge University Press, 1998.

[46] E. Wenger, 'Communities of Practice: The Key to Knowledge Strategy', in *Knowledge and Communities*, eds., E.L. Lesser, M.A. Fontaine, and J.A. Slusher, pp. 3–20. Butterworth-Heinemann, (2000). first published 1999.

[47] P. Zave and M. Jackson, 'Four Dark Corners of Requirements Engineering', *ACM Transactions on Software Engineering and Methodology*, **6**(1), 1–30, (January 1997).

The PROMOTE® approach: Modelling Knowledge Management Processes to describe an organisational KMS

Dimitris Karagiannis[1] and **Robert Woitsch**[2]

Abstract. This article introduces the EC-Project PROMOTE® (IST-1999-11658) [1], [2], [3], where an overall framework for process-oriented knowledge management is being developed. The focus of the project is to introduce a modelling language that is sufficient to describe the organisational memories and implement a Knowledge Management System (KMS). These models are seen as an overall management view that is tool and method independent. The Use Case "Software Development" where a software development process is supported to enhance quality is introduced, example models are depicted and the realisation concept is pointed out. The evaluation of knowledge management is briefly mentioned by introducing a Balanced Scorecard model that has been adapted to the needs of knowledge management.

1. INTRODUCTION

There is a significant gap between the importance of knowledge management and the realisation on all levels in an organisation: There are many surveys that show that knowledge management is recognized as a management task with high priority. When looking at concrete projects and initiatives, however, knowledge management receives much less attraction. Lack of time is a main reason that knowledge workers mention when asked why they do not support knowledge management.

A possible reason for this gap between necessity and reality is separation of knowledge management from the core business. Another reason is the difficulty to access available knowledge. Identifying an expert or finding documents with relevant information is a time consuming and often frustrating task. Even worse, people often are not aware that helpful knowledge or information might be available.

To overcome these barriers the PROMOTE® approach provides a solution to two critical challenges of knowledge management

- integration with the operational business: knowledge management tasks are associated with activities in business processes.
- providing access to available knowledge: explicit graphical knowledge structures help to get an immediate overview of available knowledge - people with required experiences - and information - codified explicit knowledge.

Knowledge management consists of many subtasks like identification, access, storage, use, distribution, etc. From these the use of knowledge is the most important. Why should vast amounts of lessons learned be stored in a service database if the service agents do not access it? What does it help to distribute experiences of successes and failures if the workers do not remember them when a new problem arises? What does it help to store product specifications if a technician developing a new product does not recognize the analogy to a similar solution?

An important challenge for using knowledge is to asses the relevance for an actual task. Knowledge is relevant if it helps to solve the problem at hand. The problem can be characterized by two criteria:

- The knowledge content: It is an obvious distinction whether we must calculate the premium of a life insurance, fix the interest rates of a mortgage or diagnose the error in a defect computer device.
- The work context: The work context consists of the overall process and the persons involved. Activities in general are part of a business process; the information gathered and decisions made in preceding activities of the process have a significant influence on the relevance of knowledge. For instance, the premium of a life insurance depends on the medical risk assessment.

[1] University of Vienna, Institute for Computer Science and Business Informatics, Dept. of Knowledge Engineering, Bruenner Strasse 72, A-1210 Vienna, Austria, dk@dke.univie.ac.at

[2] BOC-ITC Ltd., Bäckerstrasse 5/3, 1010 Vienna, Austria, promote@boc-eu.com

PROMOTE® [1] is a model-based and IT-based approach to knowledge management using the concept of an organisational memory information system to store relevant information and provide pointers to people with relevant know how.

The benefits of using a model based knowledge management approach like PROMOTE® are listed as follows:

- On the base of business processes, knowledge intensive activities that strongly influence time, quality and cost of an process are easily identified and supported.
- The integration of knowledge models with business process models, and evaluation models supports an overall management view with consistent analysis, evaluation and coordination.
- The definition of an knowledge management approach by knowledge models is tool and method independent. To realise the approach several different knowledge management tools are able to be combined.
- With evaluation models like Balanced Score Card Models it is possible to evaluate the knowledge management approaches, successful approaches are able to be distributed through knowledge model documentation.

PROMOTE® supports various model types to deal with the above-mentioned content and context characteristics:

- Topic maps are knowledge structures that model the knowledge content. They are semantic networks consisting of knowledge objects (topics) and relations between them. A graphical representation of topic maps helps a knowledge seeker to navigate in the organisational memory: If, for instance, he is looking for knowledge about cancer, a medical topic map shows all the related topics like smoking etc. Thus the knowledge seeker gets hints about relevant knowledge he/she did not think of.
- Skill models relate topic maps to people and represent the skill status of a particular person with respect to topics in a topic map.
- Process models represent the work context. Knowledge objects and people can be associated to knowledge-intensive activities via so-called knowledge processes.

The PROMOTE® approach has been developed in an EU-funded project having the same name. It has successfully been applied in two trial cases. The following section gives an overview of the knowledge management methodology of PROMOTE® and the application within a test scenario.

2. PROMOTE METHOD TO DEVELOP AN ORGANISATIONAL KMS:

This section describes the previously mentioned access to available knowledge. The PROMOTE® approach uses the business process as a starting point of the knowledge management approach as this process is not only seen as “a set of manual, semi-automatic or automatic activities, that are executed under the restriction of certain rules to achieve an organisational goal” (translated) [4], but also as the Know-How-Platform of an organisation, that will be realised by value chains to achieve the strategic goals of an organisation. Supporting the critical tasks of business processes automatically leads to a knowledge management approach that assists users in their daily work and therefore directly focuses on the operational knowledge. The authors are well aware that there exists several definitions of knowledge and knowledge management, within PROMOTE knowledge is seen as “humanised information” [3]. PROMOTE has therefore the aim to support users at critical tasks with information in a way the user can interpret this information. This information exchange will be defined by so-called “knowledge management processes” (KMPs) that define the building-, identification-, access-, storage-, distribution-, and evaluation-process of an organisational memory.

In PROMOTE these KMPs are seen as important, as the interaction between users and the organisational memory can be defined, distributed and evaluated. The KMP-categories used in PROMOTE are described as following:

- **Knowledge model building processes:**
  This category of knowledge management processes includes the analysis of business processes, the modelling of knowledge models and the validation of knowledge models.
- **Knowledge identification processes:**
  This category of knowledge management processes includes the identification of critical business processes, the analysis of skills and competences, and the analysis of the business processes supported by knowledge management processes.
- **Knowledge access processes:**
  This category of knowledge management processes includes the interactions between human knowledge workers and the organisational memory as well as the interactions with the internet.
- **Knowledge storage processes:**
  This category of knowledge management processes includes the storage of micro articles, the categorisation of documents and the description of knowledge resources with textual annotation.

- **Knowledge distribution processes:**
  This category of knowledge management processes includes the co-ordinated generation, validation and distribution of new entries in the organisational memory.
- **Knowledge evaluation processes:**
  This category of knowledge management processes includes the definition of knowledge evaluation criteria, the modelling of such evaluation criteria and the monitoring of knowledge management processes according to the defined criteria.

Each of these KMPs categories has a different effect on the organisational memory. The definition of strategic knowledge goals help to focus on the right category. The next step is to define tools to support the selected KMPs. In today's literature there are many approaches to classify knowledge management tools, within PROMOTE these tools have been mapped to the previously mentioned KMPs.

Table 1 gives an overview how KMP-categories could be mapped to KM-tools.

Table 1 KMP categories and KM-tool mapping

	Tool Mapping
Builder	Workshops, Questionnaires, Interviews
Identification	Analysing tools, Knowledge maps, Yellow Pages, Expert reviews, Skill-games
Access	Quality circles, Project organisations, Communication platforms, Virtual teams, Distributed Project teams, Groupware, Discussion forum, Document management systems, lessons learned databases, Frequently asked questions, Organisational memory information systems, Search and retrieval, Guidelines, checklists, Organisational handbook, Micro Articles.
Storage	Discussion forum, Document management systems, lessons learned databases, Frequently asked questions, Organisational memory information systems, Guidelines, checklists, Organisational handbook, Micro Articles, Blackboard, Data base
Distribution	Knowledge Brokers, Incentives, Groupware, E-mail, Video conference, Multimedia databases, E-Training.
Evaluation	Balanced score card

Using such mapping tables, each critical task can be supported individually depending on the problem category, the KM-strategy and on the user. To ensure an overall knowledge management framework each knowledge management approach is defined by knowledge models that enable a complete documentation, an analysis of the "overall system" and a tool independent evaluation. Describing the organisational knowledge system using knowledge management processes enables a process based analysis and a tool independent design of an overall knowledge management approach.

The main focus of PROMOTE® is therefore the design of an organisational knowledge system using knowledge models, to enable an export of this model information to external knowledge management tools. Some of the above mentioned tools like Yellow Pages, Micro Articles, and Search Engines are realised in the PROMOTE-prototype as so-called Web-Components to enable a rapid prototyping approach of a knowledge management system. For a complete scenario other tools have to be accessed via interfaces like Meta Search Engines, Databases and Document Management Systems. A Knowledge Management Cockpit can be realised by modelling evaluation models linked with the knowledge management models and generating an Evaluation Web-Component out of this model information.

The next section describes the realisation of the PROMOTE® method by introducing the PROMOTE-Knowledge model types. Realisation of the PROMOTE method to define an organisational knowledge system

This section describes the realisation of the above mentioned PROMOTE® method during the project, introducing the knowledge model types to describe an organisational memory on a model basis.

The following three axioms explain the specific PROMOTE® approach:

1. Process Based Knowledge Management as Modelling Framework:
   A model-based approach based on Process models (PM) was selected.
2. Formal Model as Knowledge Processing Framework:
   A formal model to evaluate and specify the model language was defined.
3. Meta Modelling as Conceptual Framework:
   The modelling concept is based on a Meta^2 Model.

These three axioms of PROMOTE® distinguish this approach from existing tools and methods. Knowledge modelling tools describing Mind-Maps or Topic Maps are not covering the dynamic aspects of knowledge management like the KMPs and they are not supporting the integration of Knowledge management approaches with Business processes.

Existing tools providing this integration of knowledge models and business processes suffer from lack of individualization that can be implemented through the Meta-Modelling concept. The PROMOTE® idea is to analyse the existing business processes and the existing working environment and to identify so-called "knowledge-intensive-tasks". These knowledge intensive tasks are further analysed and described using various model types to define an organisational knowledge system based on knowledge management processes.

Table 2 gives an overview of the PROMOTE®-model types and a short description of making the models operational.

Table 2 The PROMOTE® Knowledge models and their possible realisation

Model type	Description	Making models operational
Business process related model types (BPM)		
Business processes:	Definition of distributed business processes.	Business process- and Workflow Management.
Working environment:	Definition of a role-based working environment.	Business process management, Organisational handbook, HRM.
Knowledge processing model types (KPM)		
Skill documentation	Definition of competences, skills and interests.	Training concepts, Yellow Pages, Project Team Selection.
Knowledge structure	Definition of topics, keywords and semantic categories.	Search engines, Meta Search engines, Content Management.
Knowledge resource pools	Definition of knowledge resources.	Document Management, Groupware, Portals.
Knowledge process models	Definition of knowledge intensive tasks.	How-To Databases, Micro articles.
Knowledge management processes	Definition of the knowledge management processes.	Organisational KMS, Best practice databases, Realisation of Push-Technologies.
Security models	Definition of user rights and access profiles.	Portal Management, Single user login.
Workbench models	Definition of an individualized Web-portal.	Configuration of "MyPortal".
Overview model type (OVM)		
Knowledge landscape	Overview of the organisational memory.	Individualize the view of the organisational memory.
Community model	Overview of the teams within a working environment.	Groupware, Discussion forum, virtual Project teams.
Process pool model	Overview of the dynamic aspects within OM.	Visualisation of companies processes.

It has to be pointed out that for the realisation of knowledge management approaches it is not necessary to define all model types in detail. During the analysis of the knowledge management approach the used concepts are selected and the according knowledge models are defined. This procedure is briefly described in the next section introducing a Trial Case of PROMOTE®.

3. REALISATION OF THE PROMOTE PROTOTYPE AT USER TRIAL CASE

This section describes the trial scenario "Software development" of the PROMOTE project and discusses the tools and models that will be used to realise an organisational knowledge system in that area.

First the business process was defined and the critical tasks were pointed out.

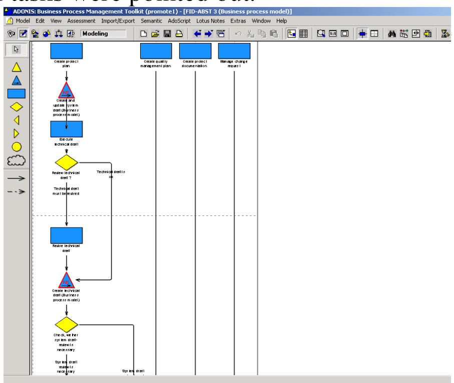


Figure 1 Screen shot of a Development process including knowledge intensive tasks

Figure 1 depicts the basic business process of the trial scenario "Software Development" where the critical tasks "Create system draft", "Create technical draft" and "Create program draft" are identified and described in more detail.

Topic maps for each of the critical task will be modelled and the necessary keywords and transformations are discussed.

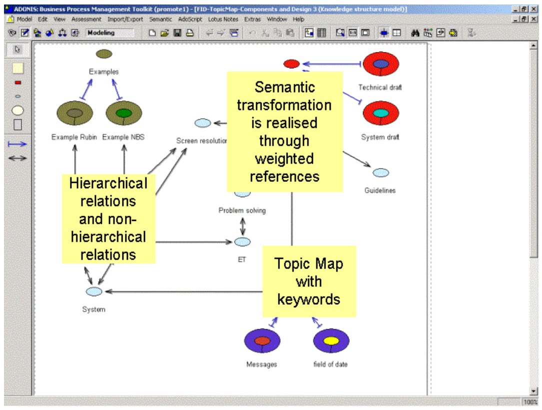


Figure 2 Topic map of the trial case

Figure 2 depicts a topic map that is realised as a knowledge structure model in PROMOTE® that has been linked to a knowledge intensive activity. One major problem of modelling semantic networks is the "knowledge transformation problem" that occurs, when departments have different views on topics.

PROMOTE solves this problem by defining several topic maps that can be linked to each other by using a “transformation link”.

These topic maps are used to define the skill profiles and the search engine interfaces. In the following, the concept of the skill documentation is briefly pointed out.

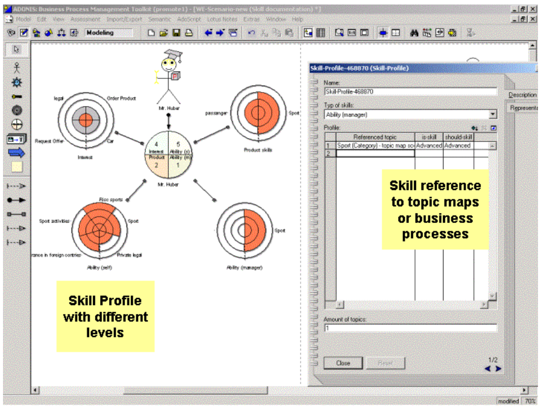


Figure 3 Skill documentation detail

Figure 3 shows a simple skill-documentation of a person. Each person has different skill-profiles that are aggregated to a so-called “Aggregated Profile”.
In PROMOTE there are the following Skill-Profile types:

- Skill Profile Interests: This Skill Profile is used to describe the interest of employees and the level the employee is interested in being trained. This profile is used to build new project teams and to document the potential of new topics.
- Skill Profile Ability (self): This Skill Profile is used to describe the abilities of employees on a voluntary basis. Users are able to enter skills they think they have. This profile is used to identify knowledge carriers and to access the knowledge of experts. This Profile is difficult to get, as many users simply do not want to document their skills. There are different ways of motivating users to keep this profile up to date.
- Skill Profile Ability (Management): This Skill Profile is used to describe the abilities of employees by the manager. The manager has the possibility to document the skills of his group by editing the skills of his employees.
- Product Skills: This skill profile reference to products of the company. Each product manager or product specialist is linked to products. This profile clearly documents the responsibility of each user.

The Skill Profiles describe the competence of either a topic (from the semantic network) or of activities within a business process. Using this framework, it is guaranteed that the skills of a person are well designed and categorised. There is also the possibility to enter “Should-” and “Is skills” at each profile. This “skill gap” has not been modelled in this trial case, as the focus of this approach was not to identify skill gaps, but to identify experts who voluntarily enter the skill documentation.

The skill documentation will be automatically imported by using existing Lotus Notes Databases.

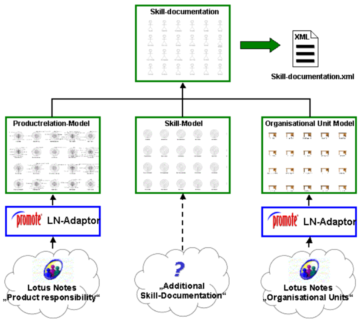


Figure 4 Automatic generated Skill documentation

Figure 4 depicts the scenario to generate this complex skill documentation using several Lotus Notes Databases. All Product Skill-Profile are generated from a Lotus Notes Database for “Product responsibilities”, the necessary information for modelling “Organisational Units” are imported from a different Lotus Notes database, finally all “Aggregated Skill Profiles” and their references are automatically generated by merging the results of the model import.

The third Skill Database is concerned with Interest Profiles and Ability Profiles and is still in the design phase. This database will be implemented during the realisation phase of PROMOTE either as another Lotus Notes database, or as an XML Database using the PROMOTE® model base.

Another concept used in this Trial scenario is a best practice database that should support users in critical decisions. PROMOTE defines the access of the database and the structure of the content of the experience base.

The content of such a database is defined with a so-called “Knowledge Process”. The authors are well aware that the terms “knowledge management process” and “knowledge process” are used differently in today’s literature but to express the PROMOTE idea, these terms are specially treated in this text:

- the Knowledge management processes defines the interaction with the organisational memory as described in section 2
- and the “Knowledge Processes” describes the content of the database. This “Knowledge Process” can be seen as a sub process of a business process, where a knowledge intensive activity (called KIT) is the “Sub-procedure call” and the “Knowledge process” is treated like a sub-process.

The reason for implementing a new model type named “Knowledge Process” and not just using a sub-process is, that additional information is needed, if an article should be generated out of such a process. The idea is to generate a short article (like a micro article [5]) out of such process models [6].

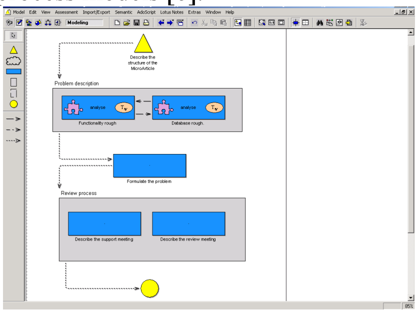


Figure 5 Definition of the structure of a short article using a „Knowledge Process"

Figure 5 depicts a definition of an article in a process-oriented manner. The start- and end-object points out that this concept can be seen as a sub-process.

The previously discussed knowledge management process define how these concepts are applied. As an example the interesting Knowledge storage management process is shown, defining the usage of a best practice database.

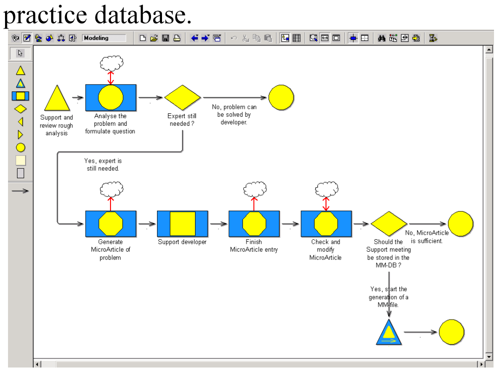


Figure 6 Example of a Knowledge Storage Management Process

Figure 6 depicts a Knowledge Storage Management Process that defines the storage of a micro article. Such a micro article should not be stored in the best practice database without the review of an expert. The above Knowledge Storage Management Process defines that the user has to suggest a consulting pool session to an expert for a specific problem. The expert can accept or deny this consulting session. If the session is accepted the review will take place and the results are stored in a best practice database.

These knowledge management processes define the interaction between users and the organisational system. The planed realisation of this user trial is mentioned in the next chapter.

4. MAKING KNOWLEDGE MODELS OPERATIONAL IN THE TRIAL SCENARIO "SOFTWARE DEVELOPMENT":

This section describes the tools that are used for the realisation of the previously mentioned knowledge management approach in the Trial scenario "Software Development". PROMOTE® has standard modules such as a Model editor, Yellow pages, Search engines, a Micro Article Generator, a Model viewer and a Knowledge management control cockpit.

In the following each concept described before is listed below and the realisation either by PROMOTE® Web-Components or by external tools is explained.

Process documentation via HTML:

The process models will be exported to HTML-Pages and can be viewed via Internet Explorer. The Processes are visualised, and descriptions and documents are attached at each critical task. Microsoft Office Documents, and Lotus Notes Databases can be accessed by clicking on the Models and by following the HTML-Links.

Meta Search Engine for Information retrieval:

A powerful information retrieval will be realised through the interaction of PROMOTE® and the U.S.U. Knowledge Miner [7]. This Meta Search engine exchanges the Topic Maps on the ISO/IEC 13250 [8] standard with PROMOTE and enables access to log data of the search engine to evaluate the tool. The search engine can be integrated in the PROMOTE Web portal through Java Servlets if appropriate.

Yellow Pages:

This concept will be realised by the PROMOTE® Web-Component called "Yellow Pages" that accesses the previously described Skill Models through the model database. The skill information can be accessed either by full text search, business processes or semantic networks through the PROMOTE® Web-Interface.

Best Practice Database:

The Best practice Database in this Trial Case is implemented as a Lotus Notes Database that stores short articles generated by the PROMOTE® Web-Component "Micro Article Generator". These short articles are defined in the models shown in Figure 5 and generated by the "Micro Article Generator" either in html or pdf format. The article are then reviewed by an expert.

Knowledge Management Process Interpreter:

The PROMOTE® portal provides a Process engine, that supports the user by starting Knowledge management processes. The knowledge management processes can be viewed in HTML. The user can start the process either as a public or as a private process through the Web-Interface. The "Tasklist" of the participating users will show that this process has been started and will

display the responsible user and the status of the process.
These concepts are planed to be implemented and evaluated during the project. The next section describes the project status and the evaluation approach of PROMOTE.

5. EVALUATION APPROACH AND PROJECT STATUS:

The PROMOTE project now finishes the implementation phase and starts the evaluation and implementation phase. The implementation of the concepts is planed to be realised according to the above mentioned scenario till summer 2002 when the project ends.

An evaluation approach will be realised to define evaluation criteria and goals that are linked to knowledge management concepts.

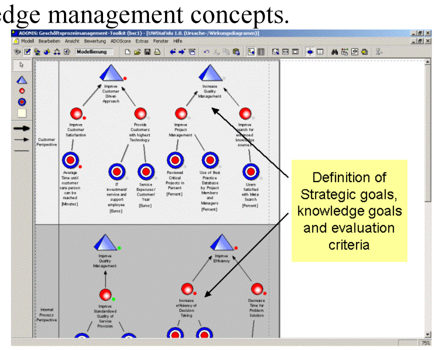


Figure 7 Knowledge management evaluation using the knowledge score card

Figure 7 depicts the evaluation of the previously described trial case using the PROMOTE® evaluation approach. PROMOTE® introduces the concept Knowledge Sore Card that is based on the Balanced Score Card and adapts this approach to the special needs of knowledge management. These evaluation models can be viewed via the HTML-component of PROMOTE® to check the performance of the organisational knowledge system.
The market launch of the first product version of PROMOTE® is planned at the beginning of 2003.

6. REFERENCES:

[1] PROMOTE 1999. http://www.boc_eu.com/promote, Access: 15.01.02.

[2] Karagiannis D., Prackwieser C., Telesko R. The PROMOTE project: Process oriented knowledge management, in Proceedings of the 3rd European Conference on Product and Process Modelling, Lisbon, 2000.

[3] Telesko R., Karagiannis D., Woitsch R. Knowledge Management Concepts and Tools: The PROMOTE Project, in Gronau N. Wissensmanagement Systeme-Anwendungen-Technologien, Shaker Verlag, Aachen 2001, p95-112

[4] Karagiannis D., The role of Workflow-Management by re-engineering of business processes (translated), in dv Management 3/94, pp 109-114.

[5] Wilke, H., Systemisches Wissensmanagement, Lucius & Lucius, Stuttgart 98

[6] Telesko R., Karagiannis D., InfoAtlas: Process-oriented knowledge management in an Intranet environment, in the proceedings of the International Symposium on Interactive and Collaborative Computing (ICC2000), Wollongong, Australia, December 11-15, 2000

[7] http://www.usu.de/Produkte/KnowledgeManagement/knowledgeminer.html
access:30.04.02

[8] ISO/IEC 13250, Topic Maps Information Technology Document Description and Processing Languages, 3.12.1999

Continuous capitalization of design knowledge

Nada Matta[1], Benoit Eynard[2], Lionel Roucoules[2], Marc Lemercier[3]

Abstract. Learning from past projects allows designers to avoid previous errors and to solve problems. Several methods have defined techniques to memorize lessons and experiences from projects in what we call project memory. This paper presents our traceability approach that allows to extract knowledge without perturbing designers' activities. Our approach is based on web technologies. In the one hand it keeps track of knowledge produced while using design tools (as a behavior model) , in the other hand, it restitutes knowledge according to a contextual situations recognition.

Keywords. Knowledge capitalization, design knowledge , project memory, product, process

1 INTRODUCTION

Knowledge management (KM), first considered as a scientist stake becomes more and more an industrial stake. It is a complex problem that can be tackled from several viewpoints: socio-organizational, financial and economical, technical, human and legal [9] . It concerns theoretical and practical know-how of groups of people in an organization. KM is defined as a continuous process of knowledge explicitation and internalization [19] .

There are two types of techniques that help to make knowledge explicit (Figure 1 .):

1. Knowledge capitalization, with which knowledge can be extracted by interviewing experts and from documents. Knowledge engineering methods are mainly used in this aim [9] .

2. Direct knowledge extraction, in which knowledge are extracted directly and dynamically from organization activity. DataMining, Textmining, tracability are some of these techniques.

For instance, some studies focus on how to keep track of an activity and especially a project. In this type of studies, the challenge is how to capitalize knowledge without perturbing actors' activities and workspace. Main questions can then arise: how to extract knowledge directly from tools and documents ? How to keep track of the issue and the evolution of a project ? How to quickly model this knowledge and represent it in a way that can be easily accessible and usable by organization actors.

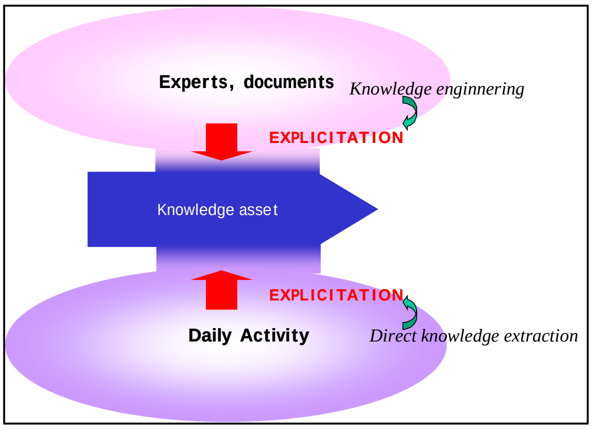


Figure 1 . Two techniques to make knowledge explicit

In this paper, we study the second type of knowledge management (direct knowledge extraction). We focus on knowledge management of a design project in order to define, what we call, design project memory (PM). A project memory can be defined as lessons and experiences from given past projects [16] . Keeping track of this knowledge can be considered as a direct extraction from several knowledge sources: documents, data bases, drawing and prototypes, meetings, activities (Figure 2 .).

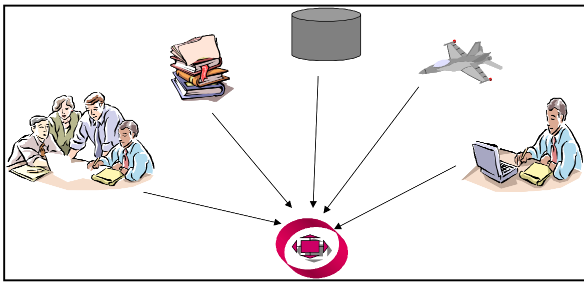


Figure 2 . Traceability of design activities

We present in this paper, traceability of engineering designer's activity. Our aim is to extract knowledge from designer's activity without perturbing him. So, we study a Web architecture that helps to define a scenario of a designer's behavior, regarding a given problem, by keeping track of used functionalities and issued information and data. Before presenting this architecture, we describe in the following section, the structure of a project memory in design.

[1]Tech-CICO, Université de Technologie de Troyes 12 rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France e-mail: nada.matta@utt.fr

[2] LASMIS, Université de Technologie de Troyes 12 rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France e-mail: {benoit.eynard, lionel.roucoules}@utt.fr

[3] LM2S, Université de Technologie de Troyes 12 rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France e-mail: marc.lemercier@utt.fr

2 DESIGN KNOWLEGDE

2.1 Knowledge modelling in design engineering

Continuous capitalisation in engineering design consists in memorising specific information that will be later on reuse in future product designs. This information is extracted from different knowledge during design process. This dynamic knowledge of the collaborative design activity is then formalised in a static project memory (Figure 3 .). The extraction and the formalisation have to be done with a maximum of transparency for designers. Thus, they would not have to manage any extra task in the design activity.

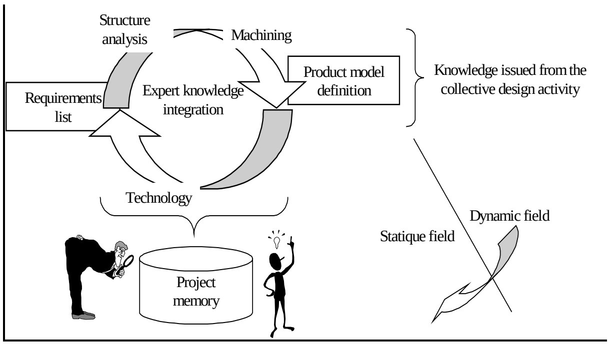


Figure 3 . Information capitalisation in engineering design.

This paper does not aim at presenting a global solution for all kinds of engineering information that must be capitalised but focuses on:

- Product data.
- Design process data.
- Design rationale data.

2.1.1 Product modelling for integrated design

Design activity is currently managed by a large group of designers that must share their points of view in order to have the product definition emerged from common decisions. Based on this Concurrent Engineering concept [26] , one goal of our research works on product and process modelling is to support the progressive product definition issued from multiple points of view knowledge integration (Figure 3 .). In other words several designers have to share their knowledge (structural analysis, technological information, machining knowledge, etc.), to define and to integrate new data on the product definition. In this way, we aim to proof that the product and particularly its geometry can be totally specified by knowledge integration from the requirements list. Thus, each data is well justified and can be really taken into account in design reuse.

Design activity is a progressive mapping of product functions to product technologies. These technologies are relating to mechanical components, machining technology, etc. According to the literature, three design phases (conceptual, embodiment and detail design) have been commonly accepted. Nevertheless, these phases are managed sequentially [20] , using axiomatic mapping [27] or concurrently [2] .

Based on an integrated design method, our product modelling tries to support strong links between functions and detailed product data [10] [22] . This model is quite similar to the mostly feature-based presented by [12] [13] or [1] . Indeed, feature presented as “a semantically endowed object that accompany product development from the customer request through to product release” [24] is very useful to define the multiple views product breakdown (cf. 2.1.2).

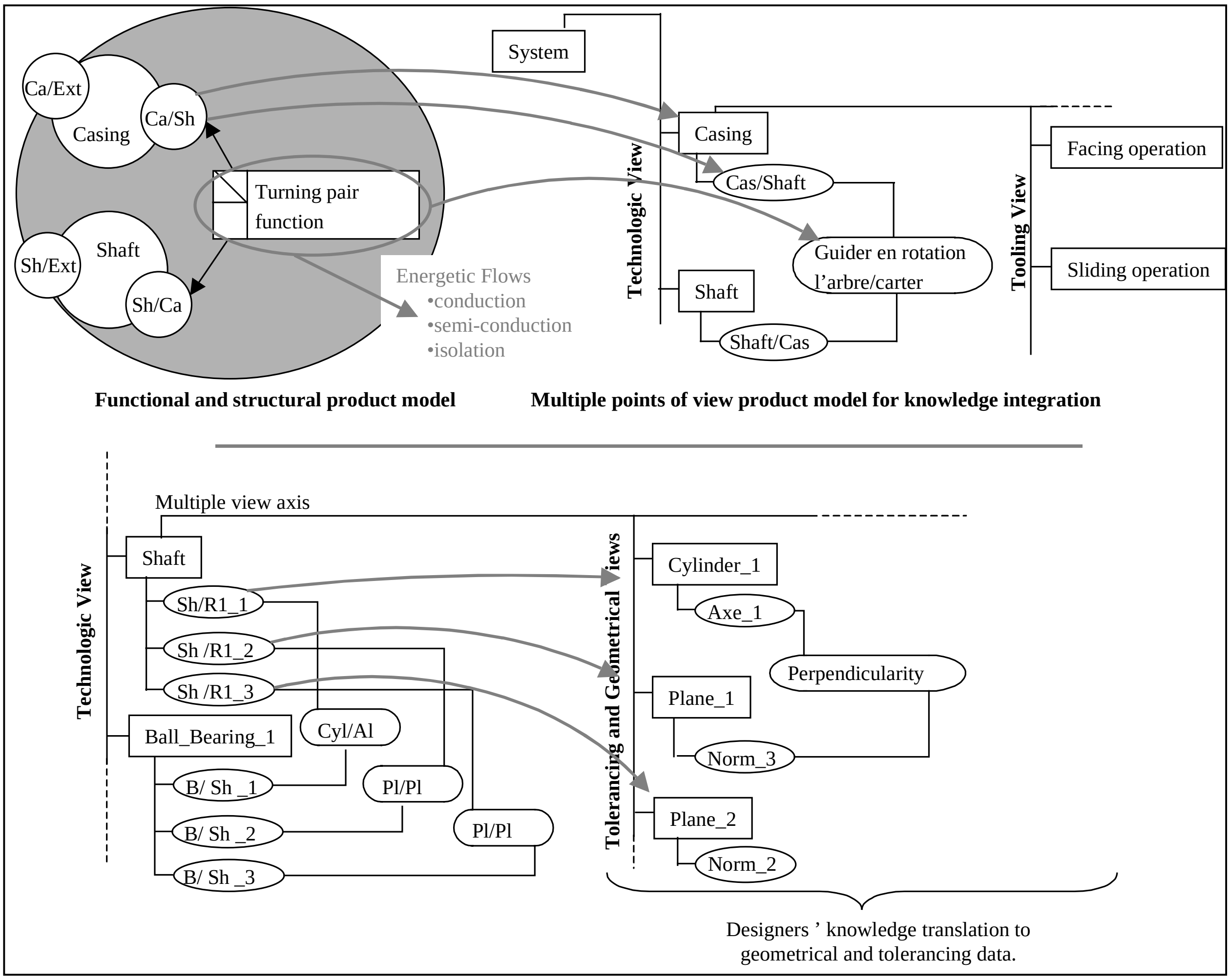


Figure 4 . Product models as data support in conceptual, embodiment and detailed design

2.1.2 A strong link between functions and structure

For conceptual and embodiment design, a function-structure model is presented. This model is a mix of several models that describe the functional and structural representations of the product. This representation is on the one hand based on bond-graph theory to treat every kind of energetic field in the product. On the other hand the representation includes graphics and rules issued from Value Engineering tools as FAST diagram (Function Analysis System Technique). This model as presented on Figure 4 is used to progressively map product functions to product structure. Each function of the FAST diagram is linked to an energetic field that is kept coherent using the bond-graph theory.

2.1.3 A multiple points of view product definition

For embodiment and detail design a model for multiple view breakdown of the product is used. These feature-based decompositions complete the product definition adding new data and new constraints from specific points of view as Machining,

Structural Analysis, etc. The model for multiple points of view is fully described in [28] . As shown on figure 4, this model represents on the one hand the structural breakdown according to the function-structure product model. This view is called the *Technologic view*. On the second hand, it is easy to create and represent new views (new decompositions) of the product (e.g.: the Tooling view).

Finally, to have the product geometry emerged, the multiple product views are translated to both tolerancing and geometric views. These two common views appear then as the result of knowledge integration. We showed in this section, how viewpoint can be useful to represent product definition. Other viewpoint representation, especially those studied [16] in knowledge representation can be used for that.

2.1.4 Computer based support for product modelling

In order to create the project memory and the continuous capitalisation (see section 3), it is necessary to manage a lot of product models. This management must also be computer-based in order to improve the transparency of the capitalisation. Therefore, extra functionality (see section 2.1.4) are added to an already-tested Co-operative Design Modeller (CoDeMo).

CoDeMo [23] has been developed to support the product modelling previously presented. It actually supports every product data that are managed via a server agent. Each designer can access and modify the product models via a client application. Computer developments of CoDeMo are based on C++ libraries provided by ILOG[4] Company. The functionality and features of CoDeMo (Figure 5 .) can be summarised as follow:

- To aid the creation of a product model using a Graphic User's Interface (GUI);
- To display the product data according to several representations (functional, geometrical…);
- To manage the database and propagate data constraints. Change notifications mean that each creation, deletion or modification are propagated from the server to every client;
- To support a Client/Server architecture in order to assist the co-operative work. The connections are currently done with RPC protocol but will be upgraded using CORBA technology.
-

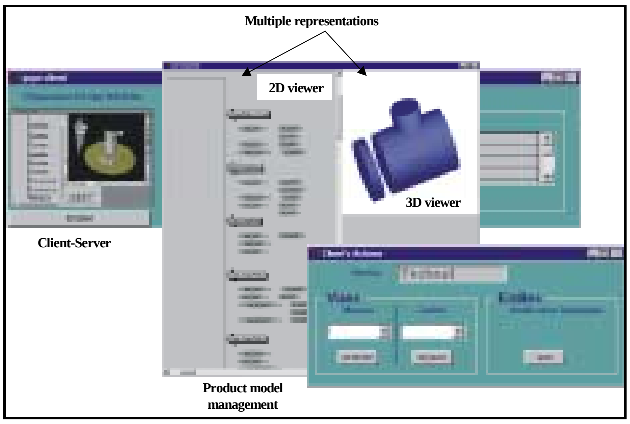


Figure 5 . Functionality of a Computer-Supported Co-operative Design Modeller.

2.1.5 Extra functionality for continuous capitalization

In the objectives of continuous capitalisation, two extra developments have been specified on CoDeMo. On the one hand (**Erreur ! Source du renvoi introuvable.**), both product and process models have to be linked. This link has to be computer supported. On the other hand it would be interesting to manage product model via XML language (see section 3).

To link product and process models would be benefit in order to manage every modification applied on the product definition. This management would step by step create an history of the product model evolution during the design process.

2.2 Modelling of Design Process

In order to have a better understanding of product development process and design activities, it is often necessary to provide details of their organisation, progress and behaviour [10] . In this section, we detail briefly various modelling languages (IDEFØ, IDEF3, Petri nets, GRAI nets and UML State Diagram) before making a rapid comparison and argue of our choice for GRAI nets.

2.2.1 Process modelling language

With IDEFØ [8] , we get a modelling language with an efficient and simple use. It provides a good graphical representation of key elements of an activity. The activity is described with a box containing an active verb characterising the activity nature. A network of arrows links the boxes and details the relationship between activities. In this relationship, activities exchanges information or objects.

IDEF3 is the issue of a research project on information integration for concurrent engineering [17] . The authors propose the description of process flow, precedence and causality relationship of activities and their logical junctions. The description of process flow uses the process flow network and is complemented with a representation of object state transition network. These two components allow to capture the behaviour and performance of process.

Petri nets [18] provide a structured description of process behaviour and allow performance assessment with associated mathematics tools. They are composed of two types of nodes: place and transition. The nodes are connected by direct arrows which specify the sequencing logic of the process. The place nodes could describe states of information or objects. The transition nodes represent operations or activities which are carried out on information or object.

GRAI nets [21] are based on three concepts: state or result, activity and support. States describe inputs and outputs (material or informational) of a transformation carried out by an activity. Activities represent operations performed between two successive states. Supports define all resources nature used by the activity. The graphical formalism could be translated in mathematical formalism thanks to the vectorial nature of states and supports : $\partial_i : (q_{i-1}, x_i) \rightarrow q_i$. GRAI nets provide specific models dedicated to discrete activity description, offering a satisfying characterisation of activity and having strong developments in terms of decision-making modelling.

Unified Modelling Language (UML) is a modelling language based on object oriented technology [7] . This language gathers the various object approaches to enable software engineering modelling. For process modelling, the UML State diagram benefits from the reference and standardised approach of object oriented technology. It

---

[4] www.ilog.com.

provides a state-event language and allows the modelling, analysis and specification of processes.

The GRAI nets combine the main quality of the previous modelling languages but require some developments in order to take into account all dimensions of engineering design. With the clarification of activity nature between states, the model benefits from logical link with the product modelling [10] . Based on the information captured in GRAI nets, we are able to represent the behaviour knowledge and process sequencing and actions of design team, etc.

2.2.2 Modelling of key elements of design process

[10] specify an extension of GRAI nets oriented to product development process modelling. He identifies three kinds of activities: design, execution and decision-making. The input and output states detail the information transformed by activities.

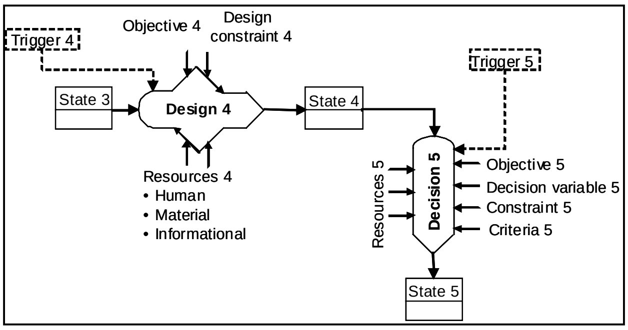


Figure 6 . Sequencing of design and decision-making activities

1- The design activity (Figure 6 .) can be defined by its iterative, creative and basically human character. It includes the understanding and analysis of problems, and the search for, creation, synthesis and proposal of solutions. The design activity is characterized by:

- the information transformed by the activity, which is represented by an input and an output state;
- the activity supports, which are of three types: material, informational and human resources;
- the specific support of the design activity, which is the design framework i.e. objectives and design constraints.

2- The execution activity is characterized by its procedural and often programmable or computational nature. It can describe the detail design of a part, the drafting of a document, etc. The execution activity is characterized by:

- the information transformed by the activity, which is represented by an input and an output state;
- the activity supports, which are of three types: material, informational and human resources.

3- As design, the decision-making activity (Figure 6 .) has a basically human character but it is purely decisional. This activity makes choices and decisions and selects alternatives in the development process. The decision-making activity is characterised by:

- the information transformed by the activity, which is represented by an input and an output state;
- the activity supports, which are of three types: material, informational and human resources;
- the specific support of the decision-making activity, which is the decision-making framework i.e. objectives, decision variables, constraints and criteria.

2.2.3 Link between product and process

Regarding the product development process, our aim is to capitalize the design history. This design history will be based on product and process modeling detailed above. It will provide a support to designers with the key elements of design project. The product dimension will be based on CoDeMo with a progressive history of product definition. The process dimension will provide a detailed description of activities, the organization and planning of the project according to [25] and [6] viewpoints.

The continuous capitalization will ensure a quick and efficient knowledge capture. The capitalization of knowledge related to product will be transparently done for designer through CoDeMo. The process modeling will provide a detailed description of transformed flow, activity support, sequencing, behavior, etc. Thus based on these three dimensions of capitalization will obtain a strong environment of capture, modeling and reuse of design knowledge.

2.3 Design rationale

Design rationale can be defined as the rationale space for problem solving. This space concerns individual and collective dimensions. Generally, discussions, alternative choices, problem solving are fleeting knowledge in a project. Nowadays the challenge is to define methods and tools in order to represent the rationale of a project and to memorize it. This type of knowledge can be characterized as:

- ❑ Problem definition: subjects, type, elements.
- ❑ Problem solving: participants, methods used and potential choices.
- ❑ Solution evaluation: rejected solutions and arguments, advantages and disadvantages.
- ❑ Decision: solution and arguments, advantages and disadvantages.

Several methods have studied how to capitalize problem solving knowledge by emphasizing the problem treated, the potential solving choices and arguments. We note for example in one hand, IBIS, QOC, DRAMA that represent the design rationale as decision space and in another hand DIPA and DRCS that suggest a problem solving modeling. Reader can have more details in [16] about these methods.

In this paper, we study relations between in one hand design rationale and in another hand, product and process models. So, we do not present design rationale capture process. For more details, see [4] .

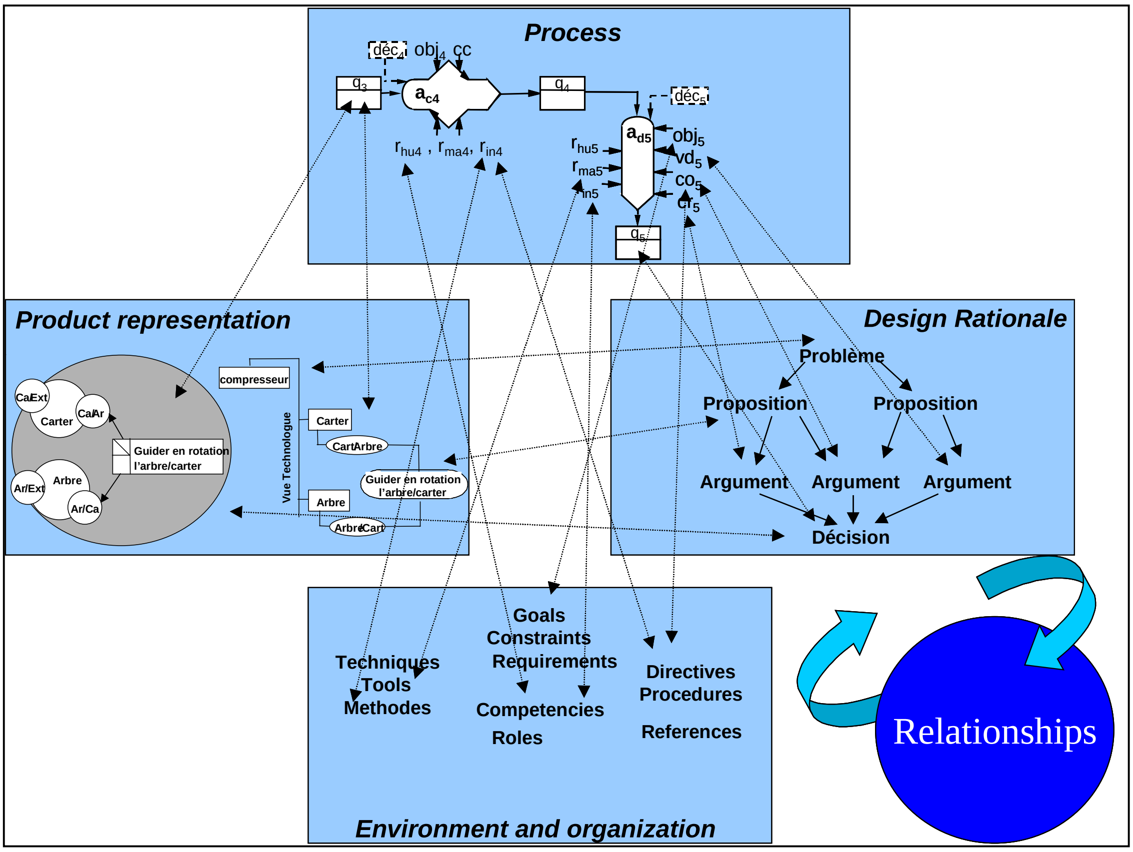


Figure 7 . Project memory structure

2.4 Structure of project memory in design

A project memory in design must consider the different part, we noted above. This type of knowledge can be organized as:

- The project organization :
  - Participants, their competencies, their roles in the project and relationships
  - Process, task organizations, constraints and requirements
- The project environment:
  - Project goal
  - References, rules, methods and directives
  - Tools and techniques
- Project realization :
  - Design rationale
  - Product description

These elements have mutual influences that is important to emphasize in a project memory (**Erreur ! Source du renvoi introuvable.**).

After presenting the different parts of a project memory, the next section describes how some of these knowledge as environment, organization, product knowledge and especially problem solving may be extracted directly from designer's activity.

3 DIRECT KNOWLEDGE CAPITALISATION FROM THE ACTIVITY

Currently, designers mostly work by using design software (ex: CAD/CAM), etc. , They even use innovation tools for creating new ideas (ex: TechOptimizer™). Our idea, is to extract the behaviour of designer by observing his activity when he uses software to solve a given problem. This behaviour can be kept as scenarios of used functions, corresponding data and documents produced, interactions (e-mails, data exchanges, …), etc. We specify a web architecture (described in the next section) that allows the observation of the designer activity [11] . XML can also be used in order to structure data extracted as a behaviour model. A knowledge engineer can then analyse behaviour models and represent environment and problem solving elements in the project memory. Figure 8 . illustrates this process.

The observation of experts' activity and problem solving has been largely used in knowledge engineering for knowledge extraction [3] . This technique is inherited from cognitive psychology and ergonomics. In this technique, the observer needs some elements related to the global project, before starting the observation. For instance, observer needs information about the step of the process the expert treats and corresponding constraints and requirements of the problem. In order to bring out these elements, the designer is first invited to identify the task he carries out when he uses software. This identification allows to establish the link between the behaviour model we observe and the project organizations and corresponding environment (design process model, actors, roles, constraints and requirements).

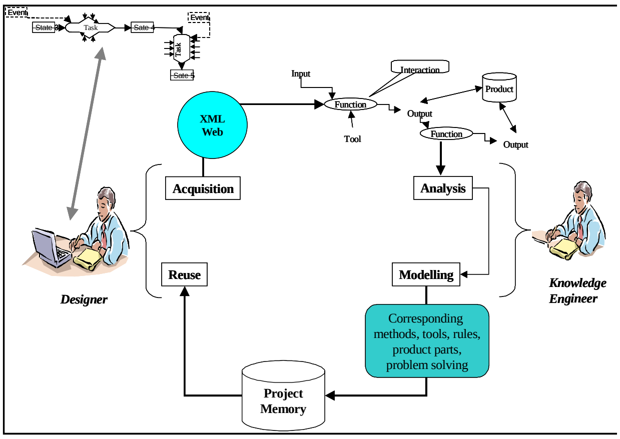


Figure 8 . Designer activity observation

We present in the following the Web architecture we defined for this aim. We show also how it can be used not only for designer's activity observation but also for knowledge restitution.

3.1 Web architecture

In this paragraph, we present the main elements of the experimental platform developed for this project. The « *project memory* » is an application localized in one place in the set of entities participating to the project. Its role consists in recovering information linked to designers' activities. These information received are heterogeneous. We have selected the XML language as the federal language.

Our project memory software is based on both XML and Web technologies. In a first version, we have favoured the Java language because it proposes efficient solutions to insure interactions with XML and Web topics [5] . To manipulate directly an XML document, the SAX interface (Simple API for XML) has been required in the XML community because it proposes an event framework. To each step of the analysis process, SAX releases an event associated to the XML element of the document. An other approach, the DOM interface (Document Object Model) has been proposed by the W3C. DOM proposes an object representation of a XML document and provides tools for the manipulation of trees. The XML document in its totality is redefined in the memory. More specifically, the JDOM API is used in the Java community. It proposes a great number of simplifications in the use of DOM by a transformation of all DOM interfaces and DOM class in real Java classes. In a Web context, the Java main proposal is the Servlet concept that has allowed the use of all Java classes in the development of complex applications linked to Web servers.

As summary, with the first version of our demonstrator, designers use a simple Web browser corresponding to Web applications localized on the central site (mail, agenda, document's transfer, …). For the technical point of view, this first version has been realized with an Apache Tomcat Web server and several Java Servlets [15] .

The version 2 of our demonstrator is still under development. However, we have already validated several elements increasing the functionality of the first version of our demonstrator. The main limitation concerns distant applications used by designers. It is indeed probable that on each site, particular applications will be used. In this case, we have to insure the information circulation to the central site. Brought solutions depend on the applications.

3.1.1 Case 1: a Web software in a distant site

A designer uses a Web application on its site. This first case is easy to manage. We modify HTML pages by adding Javascript functions. Thus, information are normally transmitted to the local Web server. After information recovery, the demonstrator broadcasts these data to the first Web server.

3.1.2 Case 2: not Web open applications

In the case of software developed for our project, it is possible to add a module of data recovery. We have implemented three approaches to insure the transfer of information to the central site. The first approach consists in an opening network connection (socket TCP/IP). We have used this solution for applications generally written in C or Pascal language. The second approach has been used for applications written in object language and especially in Java. The recovery module is a Java RMI client (Remote Method Invocation) that communicates with a RMI server localized on the central site. This RMI server is an additional element of our demonstrator. The third approach, more recent, is based on concepts of Web Services. A Web-Service is an application based on protocols of Internet that provides a specific service by respecting XML exchange format. It can also be seen as an accessible transaction by the exchange of XML documents between two sites. Web-Services represent the most promising solution for the integration of distributed services in a strongly heterogeneous context. Indeed, current solutions have some restrictions. The DCOM solution from Microsoft imposes the choice of the Windows platform. Java RMI and Java EJB (Enterprise Java Beans) support only the Java language. Finally, CORBA, the OMG solution uses only ORB. The main result research with the use of Web-Services is therefore a real interoperability of all applications. Components of Web-Services [14] are mainly SOAP (Simple Object Access Protocol), WSDL (Web Description Service Language), WSFL (Web Service Flow Language) and UDDI (Universal Description, Discovery and Integration).

3.1.3 Case 3: other cases

In the case of the use of a closed software proposed by a company, the solution consists by asking an extension of this software to be able to provide information from designer's activities.

3.2 The representation of the memory using the Web architecture

The project memory can be represented as a number of XML documents. These documents can be also linked to other data bases produced by specific product design (for instance CoDeMo) and process management tools. XML documents represent in fact, a flexible indexation of these documents. Automatic links (XLL) can be used to establish this flexible indexation and relations between all the parts of the project memory. The style sheets XSL is a good support to present the memory in different way corresponding to the needs of the user. The representation of the project memory can be illustrated Figure 9 .

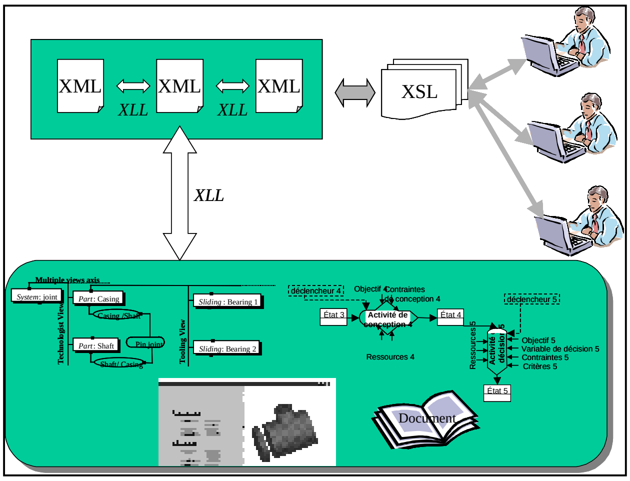


Figure 9 . A XML representation of the project memory

As we noted above, the activity observation can be also used to recognize knowledge from the memory. In fact, we plan to use a probability algorithm based on scenarios of activities in order to recognize the context of the designer and to propose a contextual access to the memory and problem solving part. The project memory can be viewed as a case base in which the environment, process and product knowledge represent the case definition and design rationale represents the case solution. So, similarity research algorithm can be used for case recognition. In project memory, the similarity can be based in different elements of the context depending on the current activity. So, the similarity algorithm must be flexible enough to support this type of recognition. Note also that some context elements can be included in the solution beside problem solving. We plan to test an algorithm based on the probability for this aim. In fact, information extracted from activity observation are used for knowledge recognition. Probability algorithm are used to compare these information with the project memory definition in order to recognize similar projects. The weight of the corresponding scenario is also incremented. So, designer can be assisted by the project memory.

4 CONCLUSION

Learning from past projects allows designers to avoid previous errors and to solve problems. A number of methods defined techniques to memorize lessons and experiences from projects. We study in this paper a traceability approach that allows to extract knowledge directly from designer's activities. The basic principle of this approach is to observe a designer facing to a problem. We use web technologies in this aim, in order to establish a behavior model of the designer by extracting and linking functions and data he uses and produces. This behavior model can be then analyzed (by the knowledge engineer) and structured in a project memory.

Our thesis is in the one hand, to keep track of knowledge without disturbing designers' activities and in the other hand, guarantee a structured and intelligent access to the memory. For that, the direct knowledge extraction as we defined, can be also used to recognize knowledge from the memory and offer a contextual restitution of knowledge. In fact, the behavior model can describe some elements of the current context and needs of the designer. These elements can be matched with the memory in order to extract similar projects that can help the designer to solve his problem. We plan to use similarity algorithm used in the Case Based Reasoning and Human Computer Interface techniques, for this aim.

In a project memory different types of knowledge must be represented: environment description, process, product and design rationale. These elements can be structured using internal and specific representation usually adopted in engineering design. The project memory can point these elements as an intelligent index based on problem solving that is the main part of traceability. With this type of representation, we do not introduce heterogeneous representation coming primly from the cognitive and artificial intelligence science "as semantic network and cognitive models".

5 REFERENCES

[1] Anderl R., Mendgen R., "Modelling with constraints: theoretical foundation and applications", Computer Aided Design, Vol. 28, n°3, pp 155-166, 1996

[2] Andreasen M.M., Hein L., "Integrated product development", Springer-Verlag, London, 1987

[3] Aussenac N. – *Conception d'une méthodologie et d'un outil d'acquisition des connaissances expertes*, PhD report of the university of Paul Sabatier, Toulouse, October, 1989.

[4] Bekhti S., Matta N., Andéol B. et Aubertin G. – Mémoire de projet : Processus dynamique de modélisation des connaissance , *Proceedings of Cooperation, Innovation and Technologies* CITE'2001, Troyes, 29-30 November 2001, p. 329-345.

[5] Bernadac J.C., Knab F., *Construire une application XML*, Editions Eyrolles, Paris, 1999.

[6] Blessing L.T.M. (1996) Design process capture and support, *2nd Workshop on Product Structuring*, Delft, The Netherlands

[7] P.A. Muller, *Modélisation objet avec UML*, Edition Eyrolles, 1997.

[8] G.J. Colquhoun, R.W. Baines, R. Crossley, A state of the art review of IDEFØ, *International Journal of Computer Integrated Manufacturing*, Vol. 6, n° 4, pp 252-264 (1993)

[9] Dieng R., Corby O., Giboin A., Golebiwska J., Matta N., Ribière M., *Méthodes et outils pour la gestion des connaissances*, Dunod., 2000.

[10] Eynard B., “Modélisation du produit et des activités de conception. Contribution à la conduite et à la traçabilité du processus d’ingénierie”, PhD thesis of the Bordeaux 1 University (France), 1999.

[11] Eynard B., Lemercier M., Matta N, Apport des technologies internet et du langage XML dans la constitution de mémoires de projet en conception de produit, *Proceedings of Cooperation, Innovation and Technologies*, CITE'2001, November 2001.

[12] Kjelberg T, Scmelkel H., “Product Modelling and Information Integrated Engineering Systems”, Annals of the CIRP, vol. 41, n°1, pp 201-204, 1992

[13] Krause F.-L., Kimura F., Kjelberg T., Lu S. C.-Y., “Product modelling”, Annals of the CIRP, vol. 42, n°2, pp 695-706, 1993.

[14] Lemnercier M., Développements Avancés de Pages Web Dynamiques , *NOTERE'97 (Colloque International sur les Nouvelles Technologies de la Répartition)*, p. 133-146, Pau, novembre 1997.

[15] Liu D., Xu W., A review of web-based product data management systems , *Computers in Industry*, Vol. 44, 2001, pp. 252-262.

[16] Matta, N., Ribière, M., Corby, O., Lewkowicz, M., et Zacklad, M. Project Memory in Design, *Industrial Knowledge Management - A Micro Level Approach*. SPRINGER-VERLAG : RAJKUMAR ROY, 2000

[17] R.J. Mayer, T.P. Cullinane, P.S. DeWitte, W.B. Knappenberger, B. Perakath, M.S. Wells, IDEF3 process description capture Method, *Information Integration for Concurrent Engineering - Compendium Methods Report*, Wright-Patterson Air Force Base, Ohio, USA (1992)

[18] T. Murata**,** Petri nets : properties, analysis and applications, *Proceedings of the IEEE*, Vol. 77, n°4, pp. 541-580, (1989)

[19] I. Nonaka, H. Takeuchi: The knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, 1995

[20] Pahl G., Beitz W., “Engineering design : a systematic approach”, Springer-Verlag, London, 1996

[21] L. Pun, Integrated discrete production control : analysis and synthesis - A view based on GRAI nets ; Elsevier, Amsterdam (1992)

[22] Roucoules L., “Méthodes et connaissances. Contribution au développement d’un environement de conception intégrée”, PhD thesis of the Institut National Polytechnique of Grenoble (France), 1999

[23] Roucoules L., Tichkiewitch S., “CoDE: a Co-operative Design Environment. A new generation of CAD systems”, CERA journal, Vol.8, n°4, pp 263-280, December 2000

[24] Shah J., “Assessment of Feature Technology”, Computer Aided Design, vol. 23, n°5, June 1991

[25] J.J. Shah, D.K. Jeon, S.D. Urban, P. Bliznakov, M.T. Rogers, “Database infrastructure for supporting engineering design histories’, Computer Aided Design, Vol. 28, n° 5, pp 347-360, 1996

[26] Sohlenius G., “Concurrent Engineering”, Annals of the CIRP, vol. 41, n°2, pp 645-655, 1992

[27] Suh N.P., “The principles of design”, Oxford University Press, New York, 1990

[28] Tichkiewitch S., “Specification on integrated design methodology using a multi-view product model”, ESDA Proceedings of the 1996 ASME System Design and Analysis Conference, PD-Vol. 80, 1996

Developing an Intranet-based Knowledge Management Framework in a Consulting Firm : A Conceptual Model and its Implementation

Reena J. Sarkar[1] and Somprakash Bandyopadhyay[2]

Abstract. Management consulting firms are considered typical examples of highly knowledge-intensive companies since they depend heavily on the expertise of their people and the nature of their assignments is knowledge-based. Hence, consulting firms have been in the forefront of thinking about how to manage knowledge However, one of the major knowledge management challenges in any organization is to develop a conceptual model to represent organizational knowledge and to use information technology for its effective implementation that would enhance right information access at right time. This paper reports our experiences in designing and developing a knowledge management framework in Management Consulting Services (MCS) of PricewaterhouseCoopers Ltd. in India (PwC India). This framework is based on a conceptual model where various knowledge sources at the content level interact to realize an integrated knowledge structure. Information technology is used here to realize an Intra-net-based framework that captures organizational structure and procedures and establishes semantic linkages among all the documents. Moreover, the framework supports sharing of informal or tacit knowledge that flows in the organization.

1 INTRODUCTION

Knowledge management refers to efforts to capture, store, and deploy organizational knowledge using a combination of information technology and business processes [1-2]. Knowledge management is a conscious strategy of getting right knowledge to right people at right time and put it into action to improve organizational performance. In recent years, knowledge management has become the terminology of many organizations in order to get competitive advantage from the efficient and effective use of their knowledge assets.

Management consulting firms are considered typical examples of highly knowledge-intensive companies since they depend heavily on the expertise of their people and the nature of their assignments is knowledge-based and mainly project-focused. They put considerable emphasis on applied creativity for solving the business problems of their clients. Their success depends on developing, selling and applying ideas to their clients. This puts heavy pressure on those firms to be innovative to meet the changing requirements of customers. Hence, consulting firms have been in the forefront of thinking about how to manage knowledge [3, 4]. KM facility can help to improve innovative culture through availability of right knowledge at right time and through knowledge sharing among the consultants. This would also avoid duplication of work, reduce learning time and improve the speed of implementation.

However, one of the major knowledge management challenges is to develop a conceptual model to represent organizational knowledge and to use information technology for its effective implementation that would enhance right information access at right time. This paper reports our experiences in designing and developing a knowledge management framework in Management Consulting Services (MCS) of PricewaterhouseCoopers Ltd. in India (PwC India). PwC is a global consulting firm and its products and services are almost exclusively based on knowledge. Hence knowledge has been placed at the center of the PwC brand: "People, Knowledge, and Worlds" and PwC is always striving to set a new standard in managing knowledge to improve organizational performance.

2 THE BACKGROUND

Knowledge management is about process, not just digital networks. Most current knowledge management activities rely on databases and Internet systems. However, few organizations have a systematic process for capturing knowledge, as distinct from capturing information. Thus, the approaches to knowledge management usually focus heavily on management of document collections viewed as knowledge repositories to be accessed in an appropriate way. The initial knowledge management practice of PwC India also relied on storage and retrieval of information from large volume of documents, often stored in logically disjoint databases within the organization. Information technology in this context provided efficient support for document management. However, finding information in a situation is too often equated with retrieving the information from those disjoint databases. Moreover, it had difficulties to meet the flexibilities demanded by knowledge-sharing approaches to knowledge management [5].

[1] PricewaterhouseCoopers Ltd, Sector V, Saltlake, Calcutta 700091 INDIA. Reena.j.sarkar@in.pwcglobal.com
[2] MIS Group, Indian Institute of Management Calcutta INDIA. somprakash@iimcal.ac.in

Existence of disjoint discussion forums or several lesson-learnt databases failed to serve their purpose. In most of the cases, the key challenge was *knowledge integration*—linking the various sources at the knowledge-content level.

If we view KM as a conscious strategy of getting right knowledge to right people at right time and put it into action to improve organizational performance, then this document-centered approach is truly inadequate. It only creates an over-abundance of information and data, not knowledge. The knowledge is not something that is "contained" in documents but it is something that can be generated or reproduced in the interaction with documents that needs to be viewed as "representations" rather than as "container"[6].

To address this issue, several researchers feel that current knowledge management practice significantly underutilizes knowledge-engineering technology [7,8]. The Knowledge engineering processes involves: using knowledge acquisition processes to capture structured knowledge systematically and using knowledge representation technology to store the knowledge, preserving important relationships that are far richer than those possible in conventional databases. In [8], three facets of the knowledge management task are considered:

- *Knowledge capture*—In the group's systematic knowledge acquisition process, a conceptual business model of the company guides case and rule capture.
- *Knowledge storage*—The group uses a knowledge representation language to codify the structured knowledge in several knowledge bases, which together make up a knowledge repository.
- *Knowledge deployment*—Through standard Web browsers on the company intranet, group members can run the knowledge bases within a knowledge server. The server answers queries far more complex than those possible with conventional database systems.

However, it is very difficult to integrate expressive reasoning tools with intranet knowledge management environments such as Lotus Notes/Domino. Therefore, it seems reasonable to conclude that while knowledge-engineering processes are ready to bring significant benefits to knowledge management projects, the implementation is still a complex task [8]. Abecker *et al.,* [9] propose an approach to build up a KM framework from existing documents to avoid employee's resistance and work disruption.

Thus, a knowledge engineering approach is needed to develop a proper conceptual modeling of the organizational knowledge in order to structure the KM process. At the same time, a simplified implementation mechanism needs to be used to ease the process of developing and augmenting the knowledge system.

With this objective in mind, a KM framework, Knowledge Point, has been designed as a single-window access to the PwC's online resources, allowing quick access to internal and external information, including resources of different strategic business units (SBU) within the organisation, industries, clients, projects and people. This framework is not a mere repository of documents but various sources at the knowledge-content level have been integrated to realize an integrated knowledge structure. Information technology is used here to realize an Intra-net-based framework that captures organizational structure and procedures and establishes semantic linkages among all the documents. Moreover, the framework supports sharing of informal or tacit knowledge that flows in the organization. The framework supports different **c**ollaborative tools so that the people can participate in different communities of interest or special interest groups to share their views and ideas and can learn from experts within the organization in an informal way.

3 A CONCEPTUAL MODEL

Acquiring and structuring corporate knowledge has proven to be the bottleneck in the design of effective knowledge systems for organizations [8]. To overcome this difficulty and to ease the work of the analyst building the KM system, the choice of knowledge structuring and the type of knowledge to capture are critical. For each domain, a body of knowledge exists and is maintained in different forms (formal and informal, structured and unstructured, as books, documents, procedures, database, etc.). The issue of what knowledge should be considered as candidate for KM system can be clarified if one distinguishes the different layers of knowledge existing in an organization and their interaction pattern.

In this context, the organization can use the knowledge-engineering process to define an organizational knowledge model—a *knowledge map* [10] — which delineates the relationships that bind the multifaceted knowledge management system at the knowledge-content level. The actual software-level bindings can use hyperlinking, or any one of a host of distributed computing techniques.

We have done it as a two-level process. First, we have identified the primary entities of PwC and their interrelationship to derive a conceptual model of the organization. An explicit conceptual model of this kind is commonly called *ontology*. This is shown in figure 1 as *enterprise ontology* [11]. At the most rudimentary level, the management consulting practice of PwC India is structured in the form of several Service Line or Strategic Business Unit (SBU), each unit specializes in providing a specific type of service to their client. Each SBU deals with projects for their clients; however, a project may require expertise from multiple SBUs. Each SBU follows a set of methods and technologies to solve the business problem for its clients. An employee of PwC normally belongs to a particular SBU. A client belongs to a particular type of industry. So the industry-specific knowledge is also required to provide service to the client. The enterprise ontology shown in fig. 1 represents this description.

Based on enterprise ontology, a knowledge network of PwC is defined in fig 2. As indicated earlier, in PwC, each SBU executes specific type of projects for a set of clients using a set of methodologies; each client belongs to a particular type of industry; people of PwC work in multiple projects, interact with multiple clients and have multiple expertise. So, there are documents related to different types of industries, different clients, lesson learnt from different projects and the methodologies followed, people involved in projects, their skill-sets and the practices and strategies of SBUs within the organization. All these documents are appropriately cross-linked so as to form a mesh-structured, continuously evolving knowledge-base of the whole organization. The basic idea is to enable users a meaningful and prompt navigation through this *knowledge network.* For example, a user viewing a project description in KnowledgePoint may be interested to know the following:

- Similar project descriptions;
- Detailed description of client for whom the project has been carried out;
- Other projects done for the same client;
- Details of the methodology / technologies used;
- Description of the industry-type where the client belongs;
- Other clients belonging to the same industry-type and the type of projects done for those clients;
- Details of people involved in the project and their skill-sets;
- People having expertise in handling similar projects using similar methodologies / technologies;
- People having expertise in similar industry-type;
- People worked with that client on different projects;

And, so on. This justifies the need for providing cross-linkage among all six entities, as shown in the figure 2.

The implementation of this conceptual model to realise a knowledge management framework will be discussed in section 5.

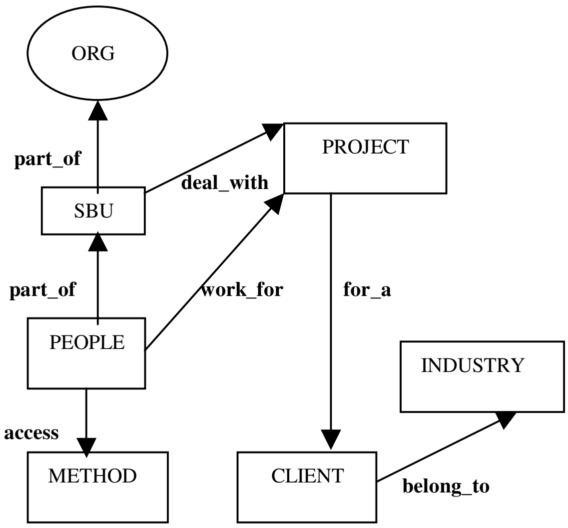


Figure 1. Enterprise Ontology of PwC

4 TECHNICAL ARCHITECTURE

Recent advancements in information technology, especially network technology, has provided a strong infrastructure for knowledge management. It enables group members to break the bonds of time and space in communicating and sharing views, ideas and experiences. An online electronic discussion forum does not require you to wait till the other person completes expressing her views. Finally, these views can be structured in an organized manner to give rise to an organizational memory. Networked computers might provide the basis for a "nervous system" that could be used to implement the capacity for organizational memory [10]. The technologies that are in use to manage knowledge include the traditional Groupware products and recent Intranet Technologies. Groupware allows the organizational record to be built in the course of everyday communication and coordination. Intranet provides the ability to organize, access and display this rich informational web.

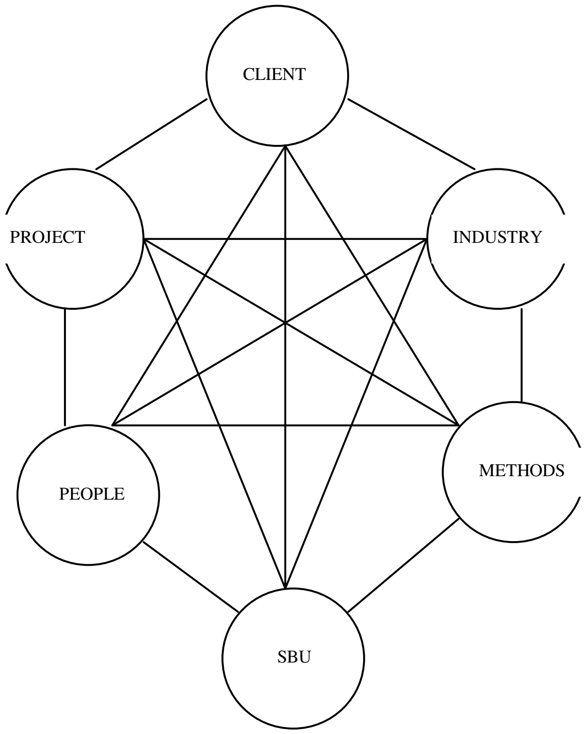


Figure 2. The Knowledge Network for PwC

Recent developments in web technology have enabled true platform independence at the client end. It also provides universal single window access to various new and legacy systems. Easy accessibility from anywhere is another important feature of the web technology. The open technology and standards of the web technology are not proprietary and we don't have to get locked to one vendor. Web solution providers are working towards incorporating as many Groupware features as possible in to their products. At the same time, Groupware vendors are also making their products web compatible. Lotus Development Corporations' Internotes 2.0 and Domino servers make the world's largest Groupware, LotusNotes, web compatible. It converts the Notes documents to HTML format on fly to make it available to a standard browser. It enables active interaction between Notes Databases & standard web browsers and supports all Internet applications, standards & protocols. Thus, two technologies are converging to one, incorporating each other's features to gain the competitive advantage.

The technical architecture is shown in fig.3. Lotus Notes Databases store information in documents about projects, clients, industry, people and other related information. All information are appropriately cross-linked using hyper-linking to generate a mesh-structured knowledge-base. Agents perform tasks which are either manual or scheduled or are initiated form the web. ASP provides authentication for user, personalization and captures the user-count and login time.

5 IMPLEMENTING KNOWLEDGEPOINT

Based on the conceptual model and technical architecture described above, a knowledge management system called KnowledgePoint has been implemented in PwC India as a single-window access to its *knowledge network*. The entry-page is shown in fig. 4. The entry-level page consists of the primary entities as depicted in the knowledge network shown in figure 2 and few other related entities. The functions associated with major entities are described below:

5.1 Client

The 'client' hotspot leads to a frame which has 2 icons: *Client and Project.* The list of clients can be also sorted by type of Industry. On clicking the 'client' hotspot, the system gives a list of all clients that have been serviced by the organization. Each client page gives the following detailed description related to that client:

- *Client description*
- *Client Financial data*
- *Client Industry and Business model*
- *Clients office network*
- *Key client contacts*
- *PwC personnel who have interacted with client*
- *Services offered to client*
- *Clients Competitors*
- *External news of client from different sources*

Each of these are linked to further information, if the reader requires to know.

To create the above profile of a new client in KnowledgePoint, there is a 'Create Client' button, which on click, gives a skeletal client profile (standard template). The user fills in the details and uses the 'Save' button on the frame to save the client profile.

From this page, there are links to other web pages such as:

- *Clients web site*
- *Parent company site*
- *Competitors web site*
- *Link to Industry page on KnowledgePoint*

Similarly, when one clicks on the Project hotspot, it gives a list of all projects listed alphabetically. They may be sorted based on the service line or SBU.

On clicking a specific 'Project', it gives the profile of the project with respects to these information:

- *Project Name and Project Code*
- *SBU* → linked to Service Line (or, SBU) page handling this project
- *Client*→ linked to client page for description of the client
- *Industry*→ linked to industry page for detailed description of the industry-type
- *Scope of the Project*→ linked to projects with similar scope
- *Brief Description*
- *Project Timelines*
- *Project Staffing* → linked to Human Resource page for details of individuals involved
- *Project Site*
- *Key Business Practice / methods / technology used → linked to knowledge repositories*
- *Key Business Practice/ methods/technology used → linked to similar projects*
- *Solution Proposed*
- *Business Benefits*
- *Non-standard solution that has been developed*
- *Lesson learnt*
- *Technical Architecture used→ linked to similar projects*
- *Project deliverable → linked to relevant databases for details*
- *Project documents → linked to relevant databases for details*

This also serves as a template while creating a new project by clicking "create project" button. Each item described above consists of a brief description against each item and linkage to other pages, whenever needed.

5.2 Service Line

The service line hotspot leads to page giving the various business units (SBUs) of PwC. On clicking a particular SBU, it may lead to a page containing different sub-SBUs under that SBU. On clicking a sub-SBU, a page containing the sub-SBU profile is presented in this format:

- *About us*
- *Methodologies* → linked to knowledge repositories for details
- *Technology* → linked to knowledge repositories for details
- *Project / Client*: List of projects and clients: completed / on-going→ linked to project/client pages
- *Staff Profile*: List of employees in the SBU with their expertise / skills /current project→ linked to Human Resource page.
- *Training*: Training courses offered → linked to Training Database and E-Learning site (a web-site for Learning and Professional Development of PwC India)
- *Knowledge Repository* → linked to Knowledge Resources page, Standardized document, templates, methodologies, Best Practice Databases

Without detailing further, we will explain very briefly the other items and their linkages.

5.3 Industry

This contains the description of different type of industries, categorized on the basis of strategic focus of PwC. Some examples are: Consumer and Industrial Products, Financial Services, Energy and Mining, Service Industry, Technology-Information-Communication and Entertainments, etc. Each category is linked to a set of sub-categories, which finally provides a list of clients under it and detailed description of the industry-type. Clients page can be accessed from this page too.

5.4 PwC India

It contains all types of corporate information with appropriate linkages to other pages, internal communications, press releases, leadership messages, global and local announcements, organizational charts, policies, service descriptions, leadership profiles, recommended knowledge resources.

5.5 Human Resources

It provides information on the staff profiles, their expertise / skill-set. Each staff is linked to individual description, SBU, project worked in, client and/or industry pages, depending on his/her expertise / experience in those areas. A user can navigate through Training Database from here to see list of courses and other training-related information. The HR page is also linked to another HR database which is used for communicating HR related news to the organization. This prevents any duplication of effort.

5.6 Knowledge Resources

The knowledge created within PwC gets translated into methodologies, policies, new tools, new methods and lessons learnt. The link ' Knowledge Resources' provides window to all this information that is the unique to the organization. It also links to the library, PwC publications and other external knowledge repositories.

5.7 Discussion Board

This is an area to share and capture informal knowledge. This type of informal interaction through discussion board can become a powerful stimulus for the collaborative development of new concepts and ideas. People can create special interest group or Communities of Practice [4]. Community of Practice can be formed in different ways :

- Certain topics that all practice members discuss and are interested in;
- Mutual engagement and binding to an entity: either a work-group within a SBU or a project;
- A shared repertoire of knowledge about a topic of mutual interest that all practice members have developed together.

Discussion Board enables open communication and knowledge sharing within the members of different Communities of Practices, created dynamically and spontaneously. Threaded discussions can be incorporated by integrating email and web functionality. A threaded discussion organizes what amounts to emails around subjects and discussions. The discussion is accessed using the web browser and the user generally starts by viewing an index of the contents in her web browser. Generally the index is organized by subject, with the primary statement listed first and the replies underneath organized by date and author. To view the content, the user selects the link. To add a response, a form is included with each message-type.

5.8 Help Desk

Help desk is a facility given to the staff to request for expertise from the SME of that subject. This results in leveraging the experience of the SME to provide instant /earliest possible solution to any query by the user/staff. This provides a support for informal learning.

6 SECURITY ISSUES

The biggest concern most executives and managers have about implementing an Intranet is security. We can make a security scheme with a lot of protection at every level, say, by means of passwords. But then we will be limiting the usage of knowledge. Security, therefore, is a continually changing balance of value, risk and practicality. The toughest part of developing a security strategy is determining what needs to be secured, and from whom. Security is not free. Every time the security level is tightened, the organization pays in terms of increased complexity of access, increased response time, and reduced communication.

Care needs to be taken that this information should not be accessible to an unauthorized user. For this purpose, login-ids have been created for all PwC employees. When he tries to login from a remote site, the proxy server asks him for its identification (unique user-id and password) and allows connection only after authenticating the user. However, not all information that is available on the content pages is for everybody. For this purpose, we have developed a privilege table. A privilege table contains a row of all the unique security classes of information and a list of all users with access to the system. The cells in the resulting table are used to record the access privileges of each user. In each cell a user either has access or does not. Privilege tables are popular because they provide a documentation format that can be easily implemented in an automated access control program. When a user logs on, the system authenticates her. When she requests access to particular information, the software looks at the privilege table to determine if she is authorized. This type of system not only simplifies the management of who gets access, but it simplifies access for the user. Because of the privilege table, the user only has to be authenticated once, rather than at each access.

7 CONCLUSION

We have described here the design and development of a knowledge management system in PwC India. This system is based on a conceptual model where various knowledge sources at the content level interact to realize an integrated knowledge network. Information technology is used here to realize an Intra-net-based framework that captures organizational structure and procedures and establishes linkages among all the documents. One of the important issues in this context is the maintenance of this knowledge network. This network resides in a dynamically changing environment and is subject to frequent changes and adaptations. The development of tools and methodologies for an efficient maintenance of this kind of network is thus a crucial research topic. In particular, we identify two major maintenance-related questions. First, the insertion of new knowledge elements into any one of the entities in the knowledge-net requires establishment of appropriate linkage with other knowledge elements in other entities. Currently, this has been done manually by knowledge management group; however, we are investigating possible approaches to create these linkage automatically, using e.g. document analysis techniques. Second, there need to be some mechanism to delete old and obsolete knowledge elements from the knowledge net. The size of the knowledge-net should not grow indefinitely. Moreover, apart from having a good technical infrastructure, it is also important to have top management commitment, appropriate culture and measurability of improvements at every stage. Hence, knowledge management is more often a managerial issue than a technical issue. In order to sustain the effectiveness of a knowledge management framework, a culture of sharing information and knowledge needs to be developed within the organization. To ensure such a culture it is important to make people realize the importance of sharing information and helping each other. Integrating KM initiatives with the organizational processes and rewarding knowledge sharing and knowledge creation are two major steps towards this direction. Formation and development of special interest groups is also one time-tested approach to initiate such cultural change. This change is essential to ensure that individuals do not reinvent the wheel within the organization. Managerial challenge in developing special interest groups is to synchronize the objectives and goals of these groups with that of the organization.

REFERENCES

[1] *Harvard Business Review on Knowledge Management*, Harvard Business School Press, Cambridge, Mass., 1998.

[2] G. Schreiber et al., *Knowledge Engineering and Management*, MIT Press, Cambridge, Mass., 2000.

[3] D. Apostolou and G.N. Mentzas, "Managing Corporate Knowledge: A Comparative Analysis of Experiences in Consuling Firms (Part I and II)", *Knowledge and Process Management*, vol. 6, No. 3-4, 1999.

[4] Sascha Uelpenich, Freimut Bodendorf, "Management of Explicit and Implicit Knowledge in Consulting Companies" , AAAI 2000 Spring Symposium on *Bringing Knowledge to Business Processes*, Stanford, CA, USA.

[5] Sorly, A., "Knowledge Sharing in Distributed Organisations", IJCAI Workshop on Knowledge Management and Organisational Memory, 31st July, 1999, Stockholm.

[6] Lueg, C. and Riedl, R., " How information Technology could benefit from Modern Approaches to Knowledge Management", Proc. of the third Int. Conference on Practical Aspects of Knowledge Management, Basel, Switzerland, Oct. 2000.

[7] N. Milton et al., "Towards a Knowledge Technology for Knowledge Management," *Int'l J.Human-Computer Studies*, vol. 51, no. 3, 1999, pp. 615–641.

[8] Alun Preece, Alan Flett, and Derek Sleeman, David Curry, Nigel Meany, and Phil Perry, "Better Knowledge Management through Knowledge Engineering", *IEEE Intelligent Systems*, January/February 2001

[9] Abecker, A.; Bernardi, A.; Hinkelmann, K.; Kuhn, O. and Sintek, M., "Towards a Well-Founded Technology for Organizational Memories". In *Artificial Intelligence in Knowledge Management.* Papers from the 1997 AAAI Spring Symposium, AAAI Press.

[10] J. Domingue and E. Motta, "Planet-Onto: From News Publishing to Integrated Knowledge Management Support," *IEEE Intelligent Systems*, May/June, 2000, pp. 26–32.

[11] Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann,Otto Kuhn, and Michael Sintek, "Towards a Technology for Organizational Memories", *IEEE Intelligent Systems*, May-June, 1998, pp. 40-48.

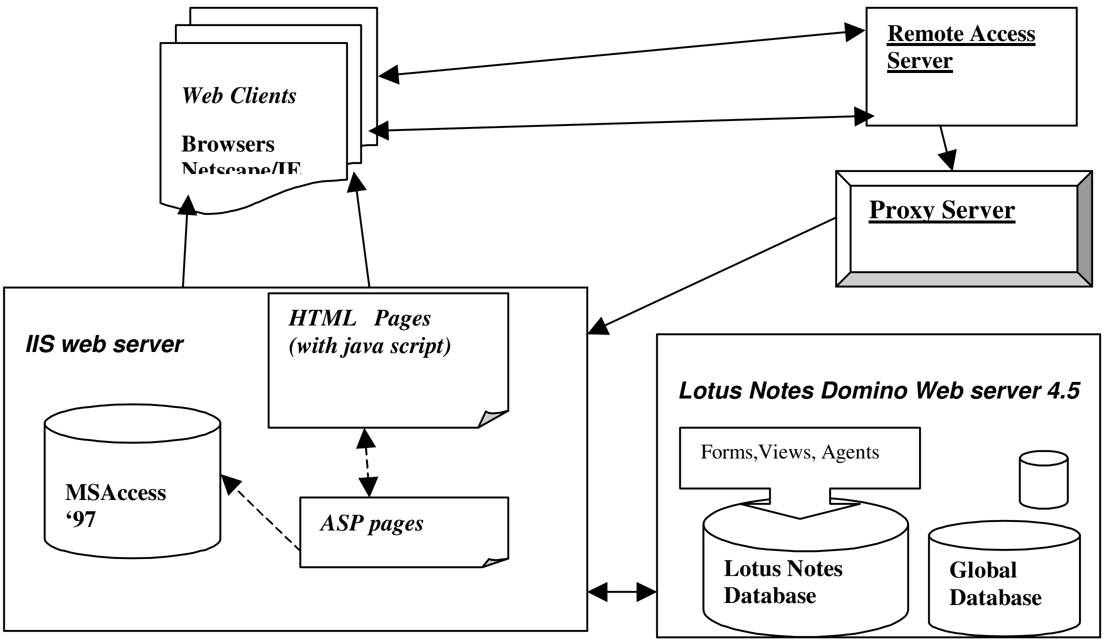


Figure 3. The Technical Architecture of Knowledge Point

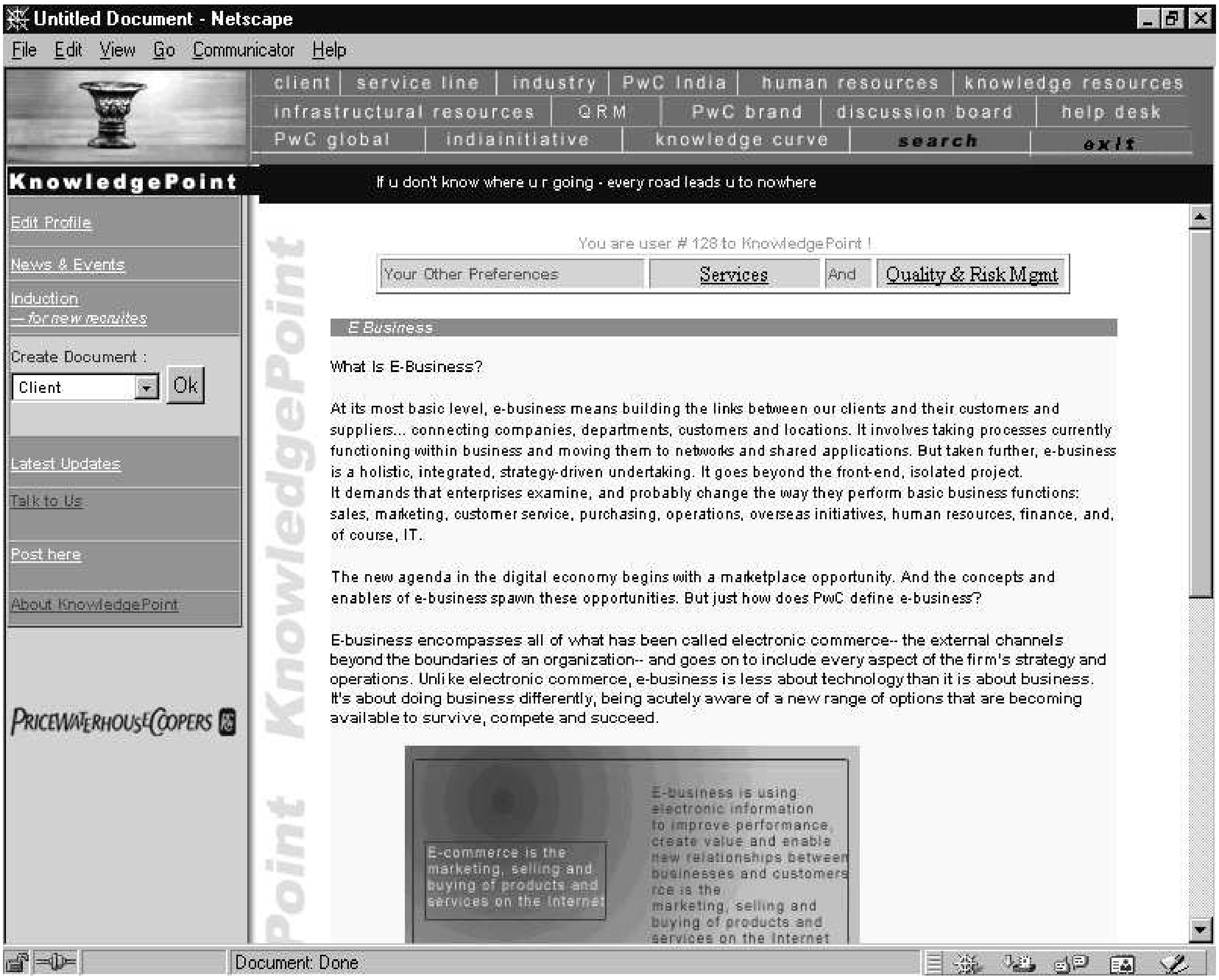


Figure 4. Entry-page of Knowledge Point

A Political Model for the Co-operative Production of Knowledge in the Design process : the Shared Medical File (SMF)

Eddie Soulier[1] and **Corinne Grenier**[2]

[1] University of Technology of Troyes (UTT), Tech-CICO Laboratory (Cooperative Technology for Innovation and Organizational Changes), email : eddie.soulier@utt.fr
[2] Groupe ESC Troyes, Tech-CICO Laboratory (Cooperative Technology for Innovation and Organizational Changes), email :cgrenier@esc-troyes.fr

Abstract. Both managerial, economic and competitive concerns in corporate practices as well as questions raised about the production of applied sciences explain the development of the vast field of research relating to sciences and the process of design which has emerged over the last ten years. The design process is complex and depends on knowledge mobilized by actors with regard to an object being produced. Existing research has primarily focused its attention on one of the three variables generally studied (knowledge, actors or object in progress), thus running the risk of divided and devalued comprehension of the whole phenomenon of design.

We offer a model (called the political model of the cooperative production of knowledge) which intends to show how what is designed is strongly dependent on the knowledge mobilized and produced by a group with various boundaries but considered as a democratic place (where *democratic* is understood in general term to describe a politicized place according to the political perspective in organization theory, thus as a place of conflict, compromise, of avoidance… [11]) where fundamental stakes around the object in production are raised. We then see the expertise as a creative political and opened-debate process of collective intelligence. We will propose an illustration of our reflection around the Shared Medical File (SMF), which represents a main but recent stake and object of interest for a sector being fully restructured.

1. THEORICAL BACKGROUND AND QUESTIONS

The innovation process involves designing and developing new products and services. The major process in innovation is the process of design and the development of objects, products or material or non-material systems. The activity of design however is still little known and the process of design remains difficult to model, particularly when we consider specific application fields. Several descriptions of the design process have been proposed. They are still too often a more or less faithful adaptation of the model of applied sciences. However, over the last few years, other approaches of design have developed which are based on the cognitive process, conversational practices, or on emerging phenomena of self-organization.

These rest on the realistic postulate that the identity of the actors involved in the design process is given at the beginning of the process and that much of the knowledge produced during the design process results from knowledge available, from characteristics of the world or constraints resulting from modeling and not from the very relative configurations of political patterns between the involved actors.

We adopt a pluralistic (or radical) perspective of organizations, by opposition to a rational or unitary perspective according to which an organization is considered as one actor with one set of coherent interests and beliefs [3].

However, the current context is characterized by a real rise in uncertainty, risks of all kinds[3] and controversies in professional knowledge [18], both in the sciences and in industry and technologies. In some fields, knowledge is passing through a crisis of legitimacy which is all the more strong since scientists in related disciplines and in so-called civil society have decided to take part in debates, thus amplifying them.

The design process is also concerned by these debates. The products of the design sciences relate to objects or systems built by human beings for human use. For this reason, the successful development of these systems involves taking into account the human aspects (dimensions) related to their design and their widespread use in society. These human aspects bring essentially into question the political dimension of the activities of design. What is political in the context of design ? It relates to what it is good and right from the point of view of all the interested parties (considering interested parties as actors who have interests to express and defend [9]). This definition is dependent on the relations of power which exist between the various actors and which become the basis for their collective and organized action. This definition is also dependent on the various representations of contexts and actions the actors mobilize during discussions and which lead to “negociated belief structure” [19] [4].

[3] This situation is related to the expansion of "biosocial " techniques (food, health, environment...) and to the extent of the associated collective risks ("mad cow", genetic engineering, pollution...) [6].

[4] According to Donnellon, Gray and Bougon [5:53], « organizational members have two alternative sets of organizing tools at their disposal : (1) shared meanings and (2) shared communication mechanisms. ». We do not adopt the position of Weick [20] for whom « sharing of beliefs is not essential to the perpetuation of interlocked behaviors » [20 : 98] according to his concept of *double interaction*. We recognize that organizational members share some commun representations (social or collective representations, [16], even if what they share is not numerous.

This power relationship is based on the respective resources, information, or formal position inside organizations [15] available to the various actors engaged in the design situation. This means on the one hand that the potential participants in the design situation are not necessarily all “actors” in the beginning; and, on the other hand, that all actors do not have the same strategic capacities given their situation. In the concrete activities of design, this takes the form of a hierarchy in the categories of knowledge and then a hierarchy in roles and status : with on one side skilled actors, who mobilize specialized, standardized, sometimes certified knowledge, and on the other side unskilled (profane) actors who take part directly or indirectly in the effort of design or who will be impacted by the object or system designed.

The design process is also dependent on what degree the group of designers is open to others. In industrial projects, this openness can take the form of taking into account manufacturers, customers and any other actor who was once excluded from the traditional approach of design (operators, sales, maintenance or after-sales staff). This is one of the stakes of converging engineering : since members inside groups use various political processes of influence so as to make the group adopt an agreement[5], how to make a success of the identification and integration of new actors to improve the process of design and its impact on the object designed?

It is on the basis of these points that this article proposes a political model of design, by raising questions on two variables which are the production of knowledge and the composition of the group, and which play a role during activities of design. According to the political metaphor, the article seeks to better understand the design of objects that we call " constitutional objects ", because they have a dual political status (sanctioning of an agreement on the basis of facts resulting from a communication process) and a cognitive status (a framing, an action plan, a representation of these facts or more precisely the representation of knowledge resulting from an epistemic process).

The aim of this article is to propose a political model of the design process around two dimensions which are fundamental for us : knowledge management and management of the collective. We adopt a managerial point of view and then wish to produce methods of assistance to the project managers and originators.

We will illustrate our modeling of the process of design using examples from software engineering, the design of information systems and a field currently under study which is the Shared Medical File (SMF) in the field of Telemedicine.

2. A POLITICAL MODEL OF THE CO-PRODUCTION OF KNOWLEDGE

A presentation of the two axes of the model (§ 2.1) will enable us to propose a schematic of this model (§ 2.2).

[5] Mintzberg [9] has identified 13 political processes (the construction of alliances, the construction of empires, the rivality beween two sides, specialized competence …) inside groups or organizations. More synthetically, Moscovici and Doise [13] propose 3 processes named conformity, normalisation and polarization. We could add another process related to enactment : adopting a 2th order definition of a problem in order to escape misunderstandings and unsuccessfull debates [21].

2.1 The two axes of the political model of design

We propose a political model for the cooperative production of knowledge, based on two axes :

- the first axis is concerned with the field and the degree of cooperation between specialists and laymen in the production of knowledge : from a simple unilateral application of universal knowledge to the cooperative formulation of what counts as a problem (problem setting).
- the second axis is concerned with the degree of structuring and legitimacy of the collectives engaged in the collective action : from the restricted team of originators producing an “enclosed” but legitimate knowledge to an extended collective uniting all the stakeholders[6], including emergent ones (external customers, trade union organizations, users, suppliers, partners…) [7].

On the axis of the production of knowledge, the principal dichotomy involves the division between ***specialists*** (or skilled people) and ***laymen***. Along this axis, the joint production of knowledge can take four distinct forms (or four situations) :

- on a first level, cooperation does not exist. The object to be designed (artifact, product, service, component, decision...) results essentially from the application of universal knowledge by the specialists[8]. The production of knowledge concerns the originators exclusively.
- on a second level, cooperation between specialists and laymen is limited to the adaptation of the object designed by universal knowledge to the particularities of the contexts of application. The originators only marginally integrate some knowledge which is specific to the needs" and use of the objects.
- on a third level, cooperation is characterized by the opening of the collective of originators to all skills and knowledge, making it possible to enrich the knowledge to be produced in the design of the object within the framework of a given problem (cooperative problem-solving). In software engineering, RAD/JAD methodologies could be classified on this level.
- on the last level finally, cooperation extends to the identification, formulation and negotiation of the problem involved in the production of the new knowledge (cooperative problem-setting). This level presupposes the construction of a

[6] The stakeholders are the individuals or the groups who depend on the organization to achieve their own goals and on which the organization also depends. The stakeholders of a firm or a project are often identified thanks to cartographies based on matrices of power/interest [10], which confirms the importance of political models of design.

[7] This axis should be related to work on the socio-dynamics of groups involved in the management of complex projects. Such work often correlates energy spent by the potential actors of a project (high, average, low) with the degree of synergy or else antagonism they are likely to express on the project. The art of project management would then involve maintaining and then widening the base of synergistic actors and controlling and circumventing antagonistic actors.

[8] Universal knowledge is a form of knowledge obtained by codification. Codification of knowledge is a conversion process of knowledge into message, wich can then be manipulated like information. Codification of knowledge is based on prerequisite of fundamentals and applided sciences. Knowledge is considered “universal” for three reasons : it is now freed from its link to a person (reification); its use is very little dependant from its context (decontextualization); its structure has in principle been optimized (rationalization). On the contrary, knowledge is considered to be “general” if it comes out of a political process of negociation rather than out of a technicist process of modelling. General knowledge deals with singularity of phenomenoms rather than search for regularity.

"space of intersubjectivity" which is not limited to the cognitive treatment of the object being designed (proposal for solutions, evaluations, goals to continue) but covers also axiological, ethical and moral dimensions. This level of cooperation results in the manufacture of general knowledge (by integration and rearticulation of local specificities) rather than universal knowledge (decontextualized and standardized).

The development of a dialogue between the various stakeholders is related to the increase in situations of uncertainty and risk. The options taken by the various groups become the subject of controversies (on the stakes, impacts, adopted solutions). These controversies involve an increasing exploration of the situation : actors and groups concerned (interest, identity, capacity...), various problems and links between them, solutions and feasible options. By integrating a plurality of points of view, requests and expectations, these controversies thus lead to the production of new knowledge through various phenomena of learning. Such a widened discussion shows that specialists and laymen and more generally each category of actor holds specific knowledge, involving diagnosis of the situation, interpretation of facts and the range of possible solutions. There is *in fine* a collective benefit which is the improvement of mutual knowledge.

On the second axis related to the structuring of collectives (or formation of groups), the main dichotomy rests on the distinction between ***instituted groups*** and ***emergent actors***. Along this axis, the joint production of the collective can also take four distinct forms :

- on a first level, the groups of design are already formed. There is no place for actors or groups of actors whose identity, functions and methods of intervention during the design have not already been perfectly defined. The stakeholders that might be concerned in fact delegate their rights of expression to these instituted representatives. In software engineering, this is typically the case of representatives of users who take part in Users Committees of the project in order to contribute to the design of the future system, to prepare its implementation, and to take part in its launch.
- on a second level, often related to the rise of controversies or dissatisfaction surrounding the design of the object, emergent groups appear whose identity, composition and borders are specified only gradually. In this phase, the essence of the difficulty for each group revolves around the constitution of a specific identity and means to be heard. In sophisticated stages of development in projects characterized by strong relational complexity, the stake rests precisely on the redefinition of the field of the actors which is no longer given, and on the comprehension of the socio-dynamics which drive them.
- on a third level, emergent groups initiate a dialogue with other emergent or already constituted groups. This third level is characterized by strong interactions and significant communication between the various groups. In terms of piloting, this stage is often most critical since it leads to the structuring of a " public opinion " whose points of view start to be articulated and which crystallize many conflicts within the process of design. This is why pilots often then begin to " deconstruct" the position of the actors by proposing for instance another formulation of the original project.
- a fourth level finally sees a new collective being born which has known how to carry out the necessary compromises and adjustments with all the stakeholders. We call these groups "extended collectives" (because of their dual sense of the variety of mobilized knowledge and the variety of interested parties taken into account); these groups are no longer limited to a mere aggregation of individuals or to already constituted groups but result from a political process of formation (in the sense of the formation of a political group).

2.2 The political model of design and the organization of collective design

We represent the political model of design with the following diagram :

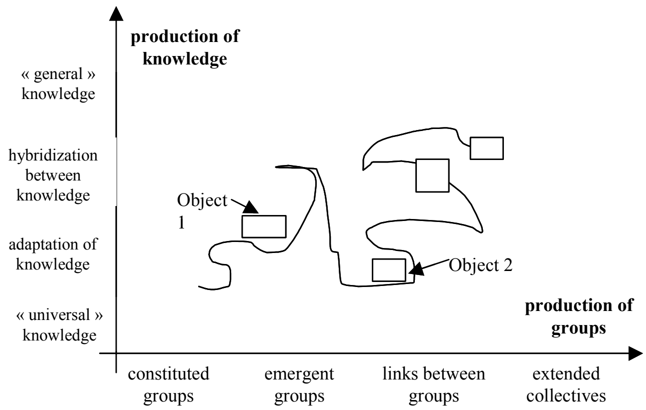


Figure 1. Political Model of Design

This model makes it possible to explore the multiple possible configurations of the process of design, keeping in mind that the two variables suggested can be analyzed both in an asynchronous and synchronous way. Thus it is possible to move along the axis of the production of knowledge without altering the modes of constitution of the groups. In the same way, it is possible to move on the axis of the composition of the groups without altering the methods of organization of the production of knowledge. The interdependence between the two variables will however be very strong in situations of design where uncertainty, risks or controversy between the stakeholders will be determining elements in the design situation.

The political path charted between the idea or the request and the finally designed object will depend on many devices conceived to better integrate the points of view of the actors involved in the design and to thus support the production of shared knowledge.

Some of these mechanisms are located at the bottom and on the left of the model whereas others, which are more participative, are on top and on the right of the model. Muller and Ali [14] have established a recent theoretical framework for the participative steps which can concern various stages of the life cycle of software.

Among the most frequent devices, we can mention benchmarking (which sometimes makes it possible to justify in advance, without debate, the choice of one data-processing solution over another), investigation of user-satisfaction, calling on experts like ergonomicists, trainers or managers in order to adapt a disfunctional system to a particular context of use, the installation of interface roles between stakeholders (correspondents, project managers-users...), the creation of new roles (like monitoring of information systems or CKO's to manage knowledge), participative techniques of design (like RAD), the direct set-up of integrated software packages of management which make it possible to implement an international professional standard without having to define the specific needs of the firm, the installation of pilot projects in order to try out a technology[9] and finally the development of levels of description (or abstraction) in the system to reduce the semantic distance between the language of the users and the conceptual language of the dataprocessing specialists (for example the hierarchy of the levels "external-conceptual-internal" in methods of design). We can also mention a significant recent trend which aims at defining governance of information systems in firms[10].

Each one of these steps presents strong points but also flaws. What is thus important is to be able to establish criteria to evaluate the various design stages.

These criteria must be consistent with the model presented, i.e. explicitly taking into account the axis of the production of knowledge and the axis of the formation and mobilization of the groups. These criteria can be structured around three dimensions :
- degree of involvement,
- level of implementation,
- induced learning.

Areas	Criteria
Degree of involvement	Intensity (participation of non-specialists)
	Opening (in terms of diversity of the consulted groups)
	Quality of contributions
Level of implementation	Technical conditions of access to the discussion
	Transparency and "traceability" of argumentative exchanges
	Clarity of the rules for organizing debates
Induced learning	Shared expertise
	Interactivity between participants

Figure 2. Procedures for the participative design

This model seeks to describe one of the dynamics at work in design processes. Its objective is to understand how to better control dynamic cooperative production of knowledge and take into account stakeholders within the activities of the design of products and services. The fact of design is seen here as a political process and design as a political activity itself aiming at producing an object as a " constitution "[11] around a dual compromise : closure / openness (groups) - universal/general (knowledge).

But process dynamics is complex, iterative, unforeseeable and all the more so since the object of the process is " something " which must pass from the status of an idea to the status of an object of work and then to a final product containing knowledge on itself and on its design context.

This object to be constructed thus also becomes an object in the process of being constructed and, as such, incorporates and crystallizes positions, divergences or agreements at critical stages in the design process. The object to be produced is thus also a constituting object of the process.

Its importance is crucial in our political model of design because we also make the assumption that this political model of design must more precisely give an account of " objects " as processes, resources and results of the cooperative activity of design at a given time. We therefore propose to call these objects "constitutional objects"[12].

3. CONSTITUTIONAL OBJECTS

We refer here in spirit to the work of S. L Star [17] on "boundary objects" where it is shown that the coordination of heterogeneous actors can be carried out thanks to the implementation of "boundary objects", which are simultaneously adaptable to various points of view and sufficiently robust to maintain their identity through them. We also integrate the work of Jeantet, Tiger, Vinck and Tichkiewitch [7] on coordination by intermediate objects in integrated teams of product design. Lastly, the contribution of E. Wenger [22] seems to us closest to the political vision we wish to explore with regard to the capacity of individuals to effectively connect their knowledge with those of others in communities of practice (cognitive synchronization).

In Wenger's work as in Star's work, connections between the various communities can be ensured by objects called " boundary objects ". All objects or artifacts which belong to several practices can play the role of boundary objects. These artifacts are seen as "reifieid" elements, which can be concrete objects (prototype, management tools, metric, version of a software, model, etc.) or symbolic systems (words of the language for example). In Wenger's

[9] It is often necessary to recreate on the "outside" (in the organization, a department...) the conditions of the environment of design ("interior") where the system was developed. This results in the installation of pilot projects, which are contexts generally furthest away from the normal operation and routines of the company, where one has gathered the most "advanced" and most desirous users of the product, where nothing is left to chance in term of training, and where the project team is most motivated. This is what explains the frequent difficulties of deployment in departments which were not pilot environments, and which can lead in certain projects to the abandoning of the installation.

[10] The governance of a company refers to the whole of its practices, structures and the procedures which specify the division of the capacity, the distribution of the responsibilities and the modes for control between the various participating components of an organization. The structure of governance establishes which interests the organization should be dedicated to and how its objectives and its priorities should be selected [8: 231-232]. The CIGREF, a french trade association representing the Management Departments of Information systems from the principal major French groups registers the "control mechanism of information systems in the strategy of the company" as the nodal point of its new associative project « CIGREF 2005" (doc. Ronéo). It is known as that "the control mechanisms of information systems raises the question of 'how the systems of information are controlled are directed' ".

[11] In the political sense of the term.

[12] Constitution ("law ", " institution"). Action to establish legally (Jur.). Way in which a thing is made up (16th century) : arrangement, composition, provision, form, organization, structure, texture. All the somatic and psychological congenital characteristics of an individual. Character, complexion, conformation, personality, temperament. " Creation " (of the world) (13th century). Action to constitute a unit; its result. Composition, construction, creation, construction, development, foundation, formation, organization. (1683) Charter, fundamental texts which determine the shape of the government of a country. Fundamental law. Constitutional : who constitutes, forms the essence of something.

work, reification indicates a process which involves giving form to an experiment by producing artifacts which solidify the experiment to some extent, at least for a time. It can take the form of an abstracted concept, tools, symbols, stories or words. Reification thus covers a great number of processes like manufacturing, design, representation, naming , description, perception, etc. Reification to some extent compensates for the contextual and evanescent character of the participation. The duality of participation/reification and its correct balance are the constituent conditions of collective practices.

For Wenger, boundary objects are characterized by four dimensions :

- *abstraction* : the general character of the boundary object leads to a certain level of abstraction.
- *versatility* : the object can be used for several activities, therefore several practices.
- *modularity* : the object consists of several parts mobilized in various situations according to the actors involved.
- *standardization* : the information contained in a boundary object must be in a directly interpretable form to be used locally.

These characteristics are relevant. However, they mainly concern mechanisms which allow for the constitution of objects, and less those concerning their use in instituted collective practices. However, what interests us in a context of design is the identification of the properties which explain the emergence, organization and functionality of such objects, rather than certain characteristics of use. If one wants to better understand the phenomena of constitution, we must propose a representation of the same criteria, but from the point of view of their genesis.

By using the theoretical background of social psychology relating to social representations ([1], [2], [12]), we propose to conceptualize constitutional objects around four variables corresponding to the variables of Wenger. We also indicate some examples of dimensions to be taken into account.

Areas	Dimensions
Structure **(abstraction)**	elements, hierarchization, dispersion of information, complexity, public dimension, focusing, autonomy of the object…
Functions **(versatility)**	interpretation, preparation for action, support for consensus, contribution to conceptualization, contribution to collaboration, contribution to argument (inferential pressure), justification of behaviours and standpoints...
Actors **(modularity)**	relationship between objects and positions, statutes and configurations of groups (individual and collective identity) and articulation with concrete social practices (concretization, anchoring...)
Normativity **(standardization)**	orientation of behaviours, legitimization, constitution and reinforcement of identity, standardization and conformisation…

Figure 3. Characterization of constitutional objects in design

4. THE SHARED MEDICAL FILE (SMF)

We will illustrate the first elements of the political model of cooperative design through the case of the Shared Medical File, which is a significant topic in the vast sector of telecare (being currently overhauled[13]). This essential object in the economy of e-health is important within the framework of our model under construction in order to question the role of new (and often challenged) actors in the process of design, and the boundary between profane and skilled knowledge.

The sector of health has been undergoing reorganization for at least 15 years now and the roles of actors and institutions have also been redefined so as to answer two major challenges : how to reconcile costs and quality ? How to handle the increasing complexity of situations and tools for diagnoses and modes of intervention and technologies for patient care ?

The sector is being reorganized mainly around the general model of the Network[14], which is presented as allowing a better control of costs, a mutualisation of expertise in favour of a more systemic approach to patient care (instead of a stepbystep approach to the patient with the risk of expensive redundancies in care or weak comprehension of disease, etc.), and especially greater autonomy for the patient, namely home-care made possible by technologies of communication (tele-monitoring, tele-diagnosis, webcam, etc.).

The Shared Medical File (SMF[15]) is one of the main elements in the implementation of a network between health partners, and for this reason it involves significant stakes : enriched medical expertise, collective and overall management of the patient, personalization of care and autonomy for the patient (who can remain at home); formalization of knowledge on patients and on medical practices, etc.[16]

[13] Telecare refers to all the applications of ICT's to the field of health and covers applications as varied as telemedicine, remote medical monitoring, teletraining, remote or collective diagnosis and all that concerns medical procedures (and pre- medical or post-medical procedures) that are computer- aided, remote, with data banks, etc. as well as electronic markets for the purchase of specific materials.... Generally, for a better knowledge of the emergent media in medical practices, see [4].

[14] Network or mode of horizontal coordination between actors; it is this term which is used to indicate the programmes of reorganization around care; we take it for granted since it is not the object of this article to define it more precisely; let us note however that there is a large variety of networks : City Hospital network for outpatient post-operative home-care in, networks of care around a particular pathology (diabetes, AIDS...) and networks of care centered on the person (network of maintenance of old people at home). This large variety has risen both from the objects of these networks as well as from the very wide variety of regulatory devices and experiments undertaken for over 20 years (when these networks were set up by associations, starting from observations on the ground and often in a largely non-formalized way).

[15] Or computerized medical File, because this last circulates more and more between the interested parties on the Internet (Intranet of hospital, extranet of a network) and more generally on the Internet or Medical Social Network (RSS designed and exploited at the request of the State by Cegetel; the RSS has been brought into service since 1998 and allows the circulation of the Electronic Files of Care between doctors and health insurance services; tools such as the Carte Vital for the patient or the Card of of Health Professionals allow a secure registering of signatures and entries on the RSS, and thus a secure registering of data relating to the Patient, under the terms of the principles on medical secrecy.

[16] The SMF can also be defined as a specific Information System around which doctors interact because they have to exchange information about the same patients. However, Information System has often been designed in accordance with the traditional hierarchical structure of hospitals and other care organizations. A more decentralized view in management and in Information System, as offered by CSCW backgroung, could improve

But the SMF is also the subject of important questions : what happens to medical secrecy, the main ethical principle in medical practices[17] or the share between private and public life ? How to ensure security of circulating or stored information ? Will it be possible to maintain the principle of continuity of care between the various components which handle their own technologies[18] ? What are the long-term costs of these information systems ?

The SMF is thus at the same time an architecture and inserted piece of knowledge which relates to the operation of the network and the patients concerned. There does not yet exist a standard model. Like any innovation in its emergent phase, one can observe an expansion of experiments (succeeding with more or less finalized SMF's) which come either from the field, or from the regulatory system, and which bring into play many actors and various carriers of different interests and stakes.

The study of this expansion shows how much the SMF being designed depends at the same time on stakeholders allowed to take part in the work of design and on their carrying scientific or profane knowledge. However its still very ambivalent status, since it calls deeply into question the sector in its entirety, also questions the productive or interesting properties of the SMF seen as a "constitutional object" in allowing the process of design to go forward.

We will develop these points in two distinct cases : the situation of design as managed by the State and as managed by various operational actors (in the field).

Experiments managed by the State reproduce the traditional diagrams of the fragmented and partitioned organization of the health sector, which is itself the object of reform in the network approach . Openness to new actors is problematic here : the patient is only too often is disregarded as a major actor while his/her needs and expectations might well be integrated in the process of design of the SMF. In fact, the patient's unskilled approaches are necessarily devalued and regarded as unscientific because produced (by definition) outside the scientific community as controlled by the State (ministries, universities, laboratories…)[19].

This is particularly important since to admit the legitimate patient as bearing knowledge could offer a springboard to many other claims, such as : what is the valorization and recognition of the role of nurses in the production and follow-up of care ? What role and responsibility is shared between the Doctor (in the broad sense), the patient and his/her family ?

Indeed, more concretely, there is the question of representation of the patient. Who, out of associations consisted assigned by the State or emerging from the field (associations of consumers for ex.) could claim to speak for the patient and his/her family ?

the quality of process design and the working of cooperation between actors.

[17] The actors, particularly the Medical Associations speak of the concept of shared medical secrecy.

[18] What is refered to here as the question of the interoperability of technologies.

[19] Certain doctors who experiment with the SMF while wanting to take into account patients note that some are not very inclined to deliver their opinion; they tend not to understand the role that is expected of them, as if it to become an active citizen were that difficult !

Debates on networks and the SMF are still too recent. The process of design runs up against the slowness of the constitution of intermediate bodies or new representative bodies in a political and professional play strongly resistant to innovations. To find the right representative body and to legitimize it in its role is not easy and can take time.

On the other hand, financial actors (Medical insurance or medical benefit funds) can see their role over-valued since they are seen as legitimizing "the network approach" recommended by the State, which is carried out in the name of cost control. Such experiments thus tend to reproduce old legitimacies and models. They remain closed to new debates relating to the patient whom they want to give greater responsibility (principle of autonomy) but no role in discussions, since no actor representing patients takes part in the design of the SMF.

Thus, this process of design internalizes social debates which should make development of the SMF an appropriate forum for a complete recasting of the health system, but which block it for the same reasons, because of their importance.

Emergent experiments from the ground also carry political questions. They often take place in partitioned and fragmented organisational contexts whose operation in networks is too recent to be widely accepted. Thus, the SMF which tends to be designed is much more the result of problems which each participant wishes to see regulated rather than the result of a vast project of reorganization of health care services.

The partitioned structure of the health system has hardly allowed the emergence of common knowledge and a common will to work in a horizontal way between internal services within an institution or between several institutions. Ignorance of the real roles of actors makes it very difficult to constitute an initial group for the design of an SMF. The risk is thus that the final SMF is a disjointed collection of hybrid pieces of knowledge which is not operational.

In the two cases rapidly approached, contributors of technology (ICT engineering...) or promoters (such as laboratories closely involved in the processes of teletraining and telemedicine) are easily able to make a place for themselves in design groups to better control cooperation and knowledge used during the discussions. Their importance is evident[20] but is exacerbated when the circumstances pointed out above prevent other actors from playing their roles.

Let us return to our model to understand the difficulties of design of an SMF. In both cases of design, what causes problems is openness to various actors, to different knowledge (or the level of hybridization according to our model) and to new collectives instituted or recognized as representatives and being able to act as representatives of new interests (or the level of links between the groups). Moreover this openness does not relate to the same dimensions which characterize the SMF as a constitutional object.

Circle A represents the process initiated by the State, which is confronted with the difficulties of opening up to new groups and new forms of knowledge. Circle B represents the process initiated by operational actors, who are confronted with the difficulties of forming a universal body of knowledge while starting with hybrid knowledge.

[20] The SMF involves an essential technological component.

In the first situation of design (circle A), the difficulty rises from the near impossibility for the State to admit the hybridization of knowledge. This refusal rises doubtless from a hard vision of what is seen as the normative nature of the SMF (or up to what point the State can question through the SMF the legitimacy of health institutions , quality standards of health production, etc.). One can think that when this hybridization is allowed, openness to new representatives and contributors of knowledge will be possible.

The reverse is found in the second situation (circle B). What raises problems here is the opportunities and organisational possibilities of connections between a multitude of groups and institutions that do not know how to work together or which are unaware of themselves. Here it would seem that the critical dimension of the SMF is that relating to its functions. The degree of versatility is equal only to the degree of diversity of the participating parts. However, we have said how much the experiments evoked here are often pragmatic and are discovered only as they come up while the SMF as constitutional object is processed. What is thus missing is a project (in the sense of a teleological vision of a complex process during its own process) relating to the functions of the SMF. One can in the same way think that when this hybridization of the parts is allowed, the coherent integration of disparate bodies of knowledge will be more possible.

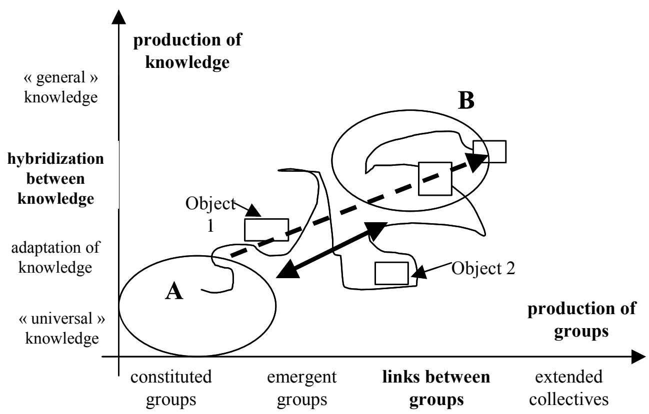


Figure 4. Spatial design of an SMF

Two particular levels of variables on each axis thus appear critical. This could militate for mixed approaches of the Up-Down and Bottom-Up type to allow learning from what emerges in each situation (full arrow connecting the two circles on the drawing).

Lastly, the movement of a mixed design process which learns from experiments initiated by the State and by instituted parts as well as those initiated by more operational parts could depend on the quality of the SMF as a constitutional object or on its structure, its functions, the actors and the degree to which it is normative.

Current experiments are still too very few to develop this point precisely.

5. CONCLUSION

Project management can take support from the political model presented in this paper. From the managerial point of view which is ours, the dialogue between cooperation and produced knowledge will interest the manager for two reasons :

- it can aim at piloting, improvement or control of a process of collaborative work and then be useful in the production of knowledge as a tool to act on cooperation;
- or on the contrary, it can aim at knowledge management or facilitate the emergence and capitalization of emergent knowledge during the design process and then act on the composition of the working group as an independent variable.

In the first case, the question is to know which knowledge to prioritize in supporting the development of cooperative work : when (in the beginning or during the process) is it necessary to introduce disorder through knowledge into a group, and would this be done for its benefit, or with the risk of blocking it, or else to even support its destruction ? Is it better to have an agreement on poor knowledge (because coming from consensus) or to promote constructive divergences ?

In the other case, the question relates to the structure of the working group. A previously defined structure, according to rational criteria of professional skills, even of political positions (in the sense of the stances of an actor) can have an economic goal (refusal of " organizational slack") or the goal of imposing order (to be pressed on a team known in advance).

But this has two weaknesses : (1) only the incidents (problems, incomprehension, tensions between the members) already known or indexed in a kind of repertory of the type "good practices" or "guide of the procedures" will be accepted and then handled[21]; (2) this mobilization of knowledge makes it difficult to bring out new knowledge.

To conclude on the two goals from a managerial point of view (to act on knowledge for better cooperation, or act on the group for better production of knowledge), the manager can easily be required to confront the risk of impoverishment :
- impoverishment of the knowledge produced in the name of the forced search for a consensus,
- impoverishment of cooperative work in the name of cohesion or availability of mobilized knowledge.

The question of knowing if a group involved in design must naturally seek a consensus for progress would merit further development.

REFERENCES

[1] J.-C. Abric *Coopération, compétition et représentation sociale*, Cousset, Del Val, 1987
[2] J.-C. Abric *Pratiques sociales et représentations*, Paris, PUF, 1994 (ed. 2001).
[3] G. Burrell and G. Morgan, *Sociological paradigms and organizational analysis* ; Heinemann Educational Books, 1979
[4] D. Carré, J.G.Lacroix (sous la dir.), *La santé et les autoroutes de l'information*, L'harmattan, 2001.
[5] A. Donnellon, B. Gray and M. Bougon, « Communication,

[21] In other words, the actors will agree to take them into account and resolve them. It is not the appearance of a problem which counts then in the progression of a work of design but the way in which this incident can be located, and understood out of a list of already existing cases, or even transposed for discussion within the design group.

meaning and organized action » ; Administrative Science Quartely, 1986, n° 31, 43-55

[6] A. Hatchuel, ‘Agir public et conception collective : l’expertise comme processus démocratique’ in Goux-Baudiment, E. Heurgon, J. Landrieu (coord.), *Expertise, débat public : vers une intelligence collective*, Colloque de Cerisy, Editions de l’aube, 2001.

[7] A. Jeantet, H. Tiger, D. Vinck, S. Tichkiewitch, ‘La coordination par les objets dans les équipes intégrées de conception de produit’, in G. de Terssac, E. Friedberg, *Coopération et Conception*, Octares Editions, 1996.

[8] G. Johnson, H. Scholes, *Exploring Corporate Strategy*, Prentice Hall Europe,1999.

[9] H. Mintzberg, *Mintzberg on Management. Inside Our Strange World of Organization*, The Free Press, NY, 1989

[10] A. Mendelow, *Proceeding of the Second International Conference on Information Systems*, Cambridge, MA, 1981.

[11] Morgan G., Images of Organization, Sage Publications, Inc., Beverly Hilss, Califormie, 1986

[12] S. Moscovici, Note towards a description of social representations, European Journal of Social Psychology, 1988, n° 18, 211-250

[13] S. Moscovici and W. Doise, Les décisions en groupes ; in Moscovici (eds) *Psychologie sociale*, Paris PUF 1984

[14] M.J. Muller, J.D. Haslewell and T. Dayton, ’Participatory Practices in the Software Lifecycle’, in Helander and al. (Eds), *Handbook of Computer-Human Interaction*, 1997.

[15] J. Pfeffer and G. Salancik, *The external control of organizations, a resource dependence perspective*; Harper & Row Publishers, 1978

[16] L. Smircich and C. Stubbart, “Strategic management in an enacted world”, *Academy of Management Review*, 1985, n° 10/4, 724-736

[17] S.L. Star ‘The structure of ill-structured solutions : Heterogeneous problem-solving, boundary objets and distributed artificial intelligence’ in M. Huhns, L. Gasser, *Distributed artificial intelligence*, vol. 2, San Mateo, Morgan Kaufman, 1989.

[18] D.A. Schön, *The Reflexive Practicionner*. How Professionals Think in Action, USA, Basic Book Inc, 1983.

[19] Walsh and Fahey, “The role of negociated belief structures in strategy making”, *Journal of Management*, 1986, n° 3, 325-338

[20] K. Weick, *The social psychology of organizing*; Random House, 1979

[21] P. Watzlawick, J. Weakland and R. Fisch, *Change, Principles of problem formation and problem resolution*; NY Horton, 1974

[22] E. Wenger, *Communities of Practice*. Learning, Meaning, and Identity, Cambridge University Press, 1998.