Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)
von der Fakultät für Wirtschaftswissenschaften
der Universität Fridericiana zu Karlsruhe
genehmigte Dissertation.

Business-Process Oriented Knowledge Management: Concepts, Methods, and Tools

Dipl.-Inform. Andreas Abecker

Tag der mündlichen Prüfung: 01.06.2004

Referent:
Prof. Dr. Rudi Studer

Koreferenten:
Prof. Dr. Peter Knauth
Prof. Dr. Grigoris Mentzas

Logic...is the beginning of wisdom, Valeris...Not the end.

Spock, “The Undiscovered Country”

Meinen Eltern, Hilde und Arnold Abecker.

Und meinen siamesischen Zwillingsbrüdern,
Michael Sintek und Ludger van Elst.

Zusammenfassung

Informationssysteme für das Organisationsgedächtnis (OMIS) zielen auf eine umfassende Software-Unterstützung für Wissensmanagement und organisationales Lernen ab. Solche Systeme sind gekennzeichnet durch: (1) die gemeinsame Verwaltung von Wissens- und Informationsquellen unterschiedlichster Art; (2) eine möglichst nahtlose Integration der Systemdienste in existierende Oberflächenkonzepte und Arbeitsweisen; (3) das selbstständige, kontext-abhängige Anbieten von Wissensmanagement-Diensten für den Benutzer.

In der vorliegenden Dissertation werden die konzeptionellen Grundlagen für solche Systeme erarbeitet, eine generische Architektur vorgestellt, und eine prototypische Implementierung gezeigt. Die generische Systemarchitektur beruht auf der dynamischen Kopplung eines Workflow Management Systems mit ontologiebasierten Wissensmanagement-Diensten, und zwar mit Hilfe ausdrucksfähiger, ontologiebasierter Metadaten-Konzepte.

Dieser OMIS Software-Kern wird ergänzt durch ein methodengestütztes Werkzeug für Gestaltung und Einführung solcher Systeme. Die komplette Lösung wurde in Fallstudien aus dem Bereich der Verwaltung und der Gesundheitsvorsorge getestet.

Insgesamt ergibt sich ein integriertes Rahmenwerk für das Geschäftsprozessorientierte Wissensmanagement, mit aktiven Wissensmanagement-Diensten unter Berücksichtigung des dynamischen Aufgabenkontexts.

Weiterhin werden in der Arbeit vielfältige Anknüpfungspunkte für weitere Arbeiten identifiziert. Inbesondere diskutieren wir: (1) Architekturen für den Handel mit Wissensgütern, auf der Basis ausdrucksmächtiger Metadaten; (2) Agentenbasiertes Wissensmanagement; und (3) schwach strukturierten Workflow zur Unterstützung der Wissensarbeit.

Darüber hinaus werden natürlich auch verwandte, ähnliche und subsumierte Arbeiten eingehend besprochen.

Abstract

Organizational Memory Information Systems (OMIS) aim at comprehensive software support for Knowledge Management and Organizational Learning. They are characterized (1) by the confederation of manifold different forms of knowledge and information; (2) by a seamless integration with existing ways of working and tools; and (3) by pro-active, context-sensitive provision of knowledge services to the user.

In this thesis, the conceptual foundations for such a system are developed, a generic architecture is presented, and a concrete prototypical implementation is shown. The system architecture is based on the dynamic coupling of workflow enactment, ontology-based knowledge services, and comprehensive ontology-based metadata.

This OMIS software core is then complemented by a method-driven tool support for designing and introducing such systems, which was tested in three case studies in public administration and in the healthcare area.

Altogether, this leads to an integrated framework for Business-Process Oriented Knowledge Management, with proactive knowledge services, respecting dynamic task contexts.

Furthermore, the thesis identifies a number of promising areas for future work which were stimulated by the presented approach. In particular, we discuss: (1) Knowledge Trading architectures on the basis of expressive metadata; (2) Agent-Mediated Knowledge Management; and (3) Weakly-structured workflow for knowledge-intensive processes.

Finally, the thesis gives a comprehensive overview of related, similar, and subsumed approaches.

Wissensmanagement ist ein typisches Beispiel für ein ***sinnvolles Thema****, das schon zu* ***20%*** *gelöst ein Unternehmen wirklich voranbringt, das aber durch den Versuch* ***der 100%-igen*** *Lösung unberechtigterweise den Ruf der völligen* ***Nutzlosigkeit*** *erworben hat.* ☺

Bernhard Kölmel, CAS AG

Diese Arbeit befasst sich mit dem Versuch, den Weg zu 100%igen Lösungen aufzuzeigen. ☺ ☺

Andreas Abecker, FZI

Table of Contents

List of Figures

List of Tables

List of Abbreviations, Tools, Institutions, and Project Acronyms

ACCI	**The Athens Chamber of Commerce and Industry,** a case study partner in the INKASS Knowledge Trading project.
AIAI	**Artificial Intelligence Applications Institute,** Edinburg, developed the Enterprise Ontology [Uschold et al., 1998].
AL	**Application Layer,** a part of the KnowMore generic OMIS architecture, see Subsection 3.1.
ADONIS	An advanced Business Process Modelling and Management tool, developed by BOC GmbH; based on a meta-modelling approach; used and further developed in a series of European research projects, such as PROMOTE and ADVISOR [Junginger et al., 2000].Was used in the KnowMore project.
AMKM	**Agent-Mediated Knowledge Management,** the idea of using analysis and design concepts, as well as software tools, from the area of multi-agent systems for building distributed KM systems, cp. Section 5.2 & [Elst & Abecker, 2004].
ARIS	**Architektur integrierter Informationssysteme,** a widespread consulting concept and modelling framework for Business Process Management; was an input for the DECOR method (Section 4.3); cp. [Scheer, 2001].
BPOKM	**Business-Process Oriented Knowledge Management,** .the idea of intertwining – for system analysis and process design, and for software support – the concepts of Business Process Management and Knowledge Management, cp. Chapter 4 and [Abecker et al., 2002].
CBR	**Case-Based Reasoning**, a technique for problem solving which looks for previous examples that are similar to the current problem. Used in several KM application areas, like Lessons Learned systems. The concept of *similarity* between complex structured objects is central for many non-trivial retrieval problems (also in an OMIS). Used in the INKASS project. Cp. [Aamodt & Plaza, 1994].
CognoVision	Now “DHC Vision”, a product for powerful management, organization, and access to manifold information and documents in the organization. Used as the core technology of the DECOR Process-Oriented Archive system. See [Müller & Herterich, 2001].
CoopIS	**Cooperative Information System,** a class of information system dealing with information from multiple sources, cp. Section 3.6.1.
CSCW	**Computer-Supported Collaborative Work,** concepts, methods, and software tools for supporting human cooperation and collaboration, in particular in the case of geographically distributed people.
DECOR	**Delivery of Context-Sensitive Organizational Knowledge,** a

	European research project about practical applications of BPOKM, see Chapter 4.
DECOR EPC	**DECOR Event-Driven Process Chain**, a slightly modified variation of the EPC approach for visual business process modelling, see Section 4.3.
DHC	**Dr. Herterich & Consultant GmbH, Saarbrücken** – a software development and consulting partner in the DECOR project; developed the CognoVision tool, coached the PVG case study.
DO	**Domain Ontology,** .formally specifies the vocabulary, concepts and relationships used by a group of agents for communicating over a given application domain.Provides attribute codomains and background knowledge for the KnowMore metadata approach. Cp. [Heijst et al., 1997] & Section 3.3.
EPSS	**Electronic Performance Support System,** a class of integrative business software systems plus associated development methodology that aims at a rigorous task-oriented efficiency improvement and training-on-the-job, cp. Subsection 3.6.1.
Frodo	**A Framework for Distributed Organizational Memory,** a bmb+f-funded German basic research project tackling, amongst other topics, AMKM and WWF issues. Cp. [Elst et al., 2004a].
IAS	**Intelligent Assistant System,** a class of software systems using mainly methods from Artificial Intelligence and Cognitive Science to enable cooperative man-machine problem solving in highly complex and dynamic application areas, cp. Subsection 3.6.1.
IDA	**Intelligent Document Access,** a software tool implementing a generic interface layer between document management and text classification systems, developed within DECOR, cp. Section 4.2.
IDEF	A set of methods for enterprise analysis and modelling, compri-sing function modelling, information modelling, data modelling, process analysis and modelling, object.oriented design, and ontology analysis and modelling. Provided the ontology engineering part of the DECOR method.
Inkass	**Intelligent Knowledge Asset Sharing and Trading,** a European research and development project aiming at an electronic platform plus associated business models for trading knowledge objects. See Section 5.1.
IKA	**Greek Social Security Institution,** a case study partner in the DECOR project. See Subsection 4.5.1 & Appendix in Chapter 7.
IO	**Information Ontology,** a formal conceptualization of the concepts, metadata attributes, their codomains, and relationships, that underly the KnowMore Knowledge Item Descriptions. Cp. [Abecker et al., 1998] & Sections 3.3 + 5.1.
IR	**Information Retrieval**, science of searching for information in documents, searching for documents themselves, searching for metadata which describe documents, or searching within stand-

	alone or networked databases, for text, sound, images or data.
ICCS	The **Institute for Computers and Communication** Systems at the National Technical University of Athens, a project partner in the KnowMore, the DECOR, and the Inkass projects.
ITIL	**IT Infrastructure Library,** an upcoming de facto standard comprising methods, documentation, and tools for managing IT infrastructures, with a focus on IT service Management.
KDL	**Knowledge Description Layer,** an element of the KnowMore generic OMIS architecture which holds ontology-based metadata descriptions for all Knowledge Objects under the management of the OMIS. See Section 3.3.
KID	**Knowledge Item Description,** a metadata set describing a concrete Knowledge Object under the control of the OMIS. KIDs instantiate concepts defined in the Information Ontology, they are stored in the KDL and processed in the KBL.
KIT	**Knowledge-Intensive Task / Knowledge-Intensive Activity,** .see Sections 0 & 4.3.
KM	**Knowledge Management,** see Chapter 1 and [Mentzas et al., 2002].
Know-Net	**Knowledge Management with Intranet Technologies,** European research and development project, see Chapter 1 and [Mentzas et al., 2002].
KnowMore	**Knowledge Management for Learning Organizations,** a German, bmb+f funded basic research project, see Chapter 2 and [Abecker et al., 1998].
KO	**Synonym for knockout:** a blow that renders the opponent unconscious. *However, in the context of this thesis, only the following meaning is relevant:*
	Knowledge Object, a tangible entity transporting knowledge, created by a knowledge asset, managed in an OMIS, cp. Section 3.4.
KOL	**Knowledge Object Layer,** an element of the KnowMore generic OMIS architecture, see Section 3.4.
mindAccess	A commercial Text Mining solution, offered by insiders Information Management GmbH, cp. 4.2.
OCRA	**Object-Centered Relational Algebra,** a knowledge representation language, developed by M. Sintek, used in the KnowMore project for ontology representation and inferencing.
OMIS	**Organizational Memory Information System,** a class of software systems aiming to support Organizational Memory, Organizational Learning, and Knowledge Management, see Chapter 1.
PLANET-EY	**Planet Ernst & Young,** Athens / GR, a Greek management

	and IT consulting house, project partner in the DECOR + INKASS projects.
Promote	**Process Oriented Methods and Tools for Knowledge Management,** a European research and development project, built on top of the ADONIS tool and methodology, see also [Hinkelmann et al., 2002].
PVG	**Plasmaverarbeitungsgesellschaft,** Springe – a case study partner in the DECOR project, see Subsection 4.5.2.
SDK	**Software Development Kit,** a programming package that enables a programmer to develop applications for a specific platform. Typically, an SDK includes one or more APIs, programming tools, and documentation.[1]
TWI	**The Welding Institute,** Cambridge / UK, a case study partner in the INKASS project. Cp. Section 5.1.
WFE	**Workflow Engine.**
WfMC	**Workflow Management Coalition,** a standardization body for workflow terminology, interfaces, etc.

[1] Definition from http://www.webopedia.com/

1 Introduction

An investment in knowledge pays the best interest.

Benjamin Franklin

Abstract: This chapter provides a motivation and overview of the work presented in this dissertation. After a brief introduction into the Knowledge Management (KM) topic and a general discussion of the role of Information and Communication technology (ICT) for KM support, we summarize the main goals and motivations and shortly present the structure of this thesis.

Preamble: *My first encounter with the multidisciplinary and holistic aspects of Knowledge Management took place in a (still existing) PhD-student network that was set up by colleagues from Prof. Warnecke's group at CIM Centrum Kaiserslautern and by colleagues from Prof. Probst's group in Geneva. In the thrilling and genial discussions with "first-generation knowledge cowboys" like Kai Romhardt, Heiko Roehl, Andreas Gissler, Gerd Stammwitz, and many others, I observed for the first time what I called later the "Product- and the Process Approach to Knowledge Management". These basic considerations went into the project proposal of the European RTD project Know-Net (Knowledge Management with Intranet Technologies) where I led the DFKI part of the project, and where I had the pleasure to collaborate with an excellent project team. In particular, I really enjoyed and learned much from the professional working style and the unresting creativity of Prof. Grigoris Mentzas. The results of this project are reported in [Mentzas et al., 2001; Mentzas et al., 2002] which were the basis for parts of this chapter. Another source of inspiration was the work with Otto Kühn who – in a remarkable manner – aimed at both taking end users serious, and nevertheless doing innovative work. Essentially, all this thesis is motivated by the user requirements coming from the industrial cases studies he did in the years before. This requirements analysis and basic conceptual work has been published as [Kühn & Abecker, 1997].*

1.1 The Role of Technology in Knowledge Management

1.1.1 Knowledge Management in a Nutshell

Since there are already numerous excellent introductions into the Knowledge Management (KM) topic (e.g., [Albrecht, 1993; Nonaka & Takeuchi, 1995; Davenport & Prusak, 1998; Probst et al., 1999; North, 1999] and many others) we just summarize some basic introductory ideas relevant for the rest of this thesis.

First, let us consider some Knowledge Management definitions found in the literature:

APQC

The American Productivity and Quality Center (APQC) outlines key KM processes and key KM enablers: *"Knowledge Management is the broad process of locating, organising, transferring, and using the information and expertise within an organisation. The overall knowledge management process is supported by four key enablers: leadership, culture, technology, and measurement."*

OVUM

The excellent and comprehensive OVUM technology report [Ovum, 1998] makes the distinction between tangible and intangible knowledge by characterizing KM as *"the task of developing and exploiting an organisation's tangible and intangible knowledge resources. Knowledge managemement covers organisational and technological issues"*.

Tom Sommerlatte

Sommerlatte's definition in [Sommerlatte, 1999] – which emphasizes the facet of goal orientation for KM – can be translated as follows: *"To acquire, process, and make accessible knowledge in a more systematic way, in order to obtain better decisions and to be better prepared for the future"*.

Conny Antoni

In the same book [Sommerlatte, 1999], we can find Antoni's definition going into the same direction (translated from German): *"identification, development, and provision of that knowledge which is relevant for the success of a company"*.

Davenport & Prusak

In their seminal book, Davenport and Prusak focus a bit more on the "management aspects" [Davenport & Prusak, 1998]: *"Knowledge Management is a formal,* *structured initiative to improve the creation, distribution, or use of knowledge in an organization. It is a formal process of turning corporate knowledge into corporate value."*

Maurer

Seen from an Artificial Intelligence (AI) perspective, Hermann Maurer adds another interesting issue, namely the person-independent storage of knowledge [Maurer, 1999]: "*Thus, the basic aim of Knowledge Management is to nurture and to increase the knowledge of individuals and to make sure that knowledge can be easily shared with others and (at least to some extent) remains even if the persons involved become unavailable."*

As one may guess from this enumeration, there are almost as many KM definitions as KM authors. Nevertheless, this collection of definitions reveals most interesting aspects relevant for a sufficiently comprehensive description of the topic. Figure 1 depicts the most important issues and facets to be taken into consideration when talking about Knowledge Management.

Manifolds facets for understanding KM

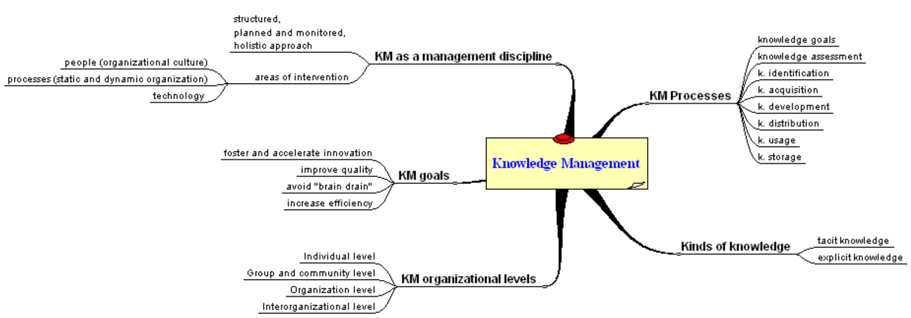


Figure 1: KM Facets

A definition which combines fairly well these different facets of the term KM can be achieved by slightly extending and adapting the one given in the University of St. Gallen's Netacademy [Netacademy, 1999]:

KM definition

Knowledge Management is a:

- structured, holistic approach
- for sustainable improvement of handling tacit and explicit knowledge (e.g., know-how, skills, notes, documentation) in an organization
- on all levels (individual, group, organization, interorganizational level)
- in order to better achieve one or more of the organization's strategic goals, like decreasing costs, improving quality, fostering innovation, increasing customer satisfaction etc.

For a more detailed explanation of this definition, please refer to [Abecker, 2004].

1.1.2 Early KM Frameworks and Approaches

A *KM framework* provides a conceptual frame of reference for ensuring completeness and integrity of a KM initiative, it typically represents a visual or conceptual tool, to set– in draft lines– the context of the specific KM approach.

Early KM frameworks

In the European research and development project Know-Net (Knowledge Management with Intranet Technologies, cp. [Mentzas et al. 2002]) we did an extensive survey of at that time existing, early KM frameworks, investigating the work by Nonaka & Takeuchi [Nonaka, 1991; Nonaka & Takeuchi, 1995], Leonard-Barton [Leonard-Barton, 1995], APQC [APQC, 1997], Romhardt & Probst [Romhardt & Probst, 1997; Probst et al., 1999], Lotus [Lotus, 2003], Angus and colleagues [Angus et al., 1998], IBM [Huang, 1997; Huang, 1998], Coopers & Lybrand [Knapp, 1998], and last but not least, the Knowledger approach of Knowledge Associates [Young, 1998].

Those inspected frameworks showed a wide range of different understandings and focal points for the KM endeavour, as it is indicated in Table 1. In particular, they fell short, however, in providing a conceptual blueprint for our comprehensive goals followed in the Know-Net Project, since they exhibited deficiencies from the point of view of:

- their "operationability";
- their completeness; and
- coverage of inter-relationships.

Actually, most of those frameworks were either too abstract (and could thus not be "operationalized"), or they were too partial or narrow, which contradicts the "holistic nature" of Knowledge Management.

Focus Area	Framework
knowledge creation	• Nonaka/Takeuchi's • Leonard-Burton's
knowledge processes	• APQC • Romhard and Probst's
technology	• Lotus • Angus and Patel
holistic	• IBM • Coopers and Lybrand • Knowledger

Table 1: Early KM Frameworks Investigated in the Know-Net Project

Hence, in order to progress with the KM discipline from the status of "vague" top management consulting without clear guidelines how to proceed for getting things running, the following aspects must be addressed:

Goals for a KM framework

1. A KM framework must be comprehensive in the sense that it covers the interdisciplinary and multi-faceted aspects of the KM idea.
2. A KM framework must be operational in the sense that it provides clear guidelines, methods, and tools (IT-based as well as not IT-based) in order to come from top-level analysis and goal setting to concrete measures and activities for implementing and operating a knowledge-based organization.
3. A KM framework must be consistent in the sense that the different parts of the framework are designed in such a way that they mutually interoperate and work together synergetically.

1.1.3 Product-centric versus Process-Centric KM

Process-centric versus Product-centric KM

A further analysis of principled approaches to approach KM showed that – seen from many perspectives – there could several times be observed a basic distinction between two separate understandings of the KM topic: the *Process-centric* and the *Product-Centric* approach [Kühn & Abecker, 1997; Mentzas et al., 2002].

1.1.3.1 Product-Centric Knowledge Management

Basic assumptions

The "product" approach implies that knowledge is a thing that can be located and manipulated as an independent object.

Proponents of this approach claim that it is possible to capture, distribute, measure and manage knowledge itself, namely by focusing on products and artefacts containing and representing knowledge; usually, this means managing manifold kinds of documents, their creation, storage, and reuse in computer-based Corporate Memories or Organisational Memory Information Systems (cp. [Abecker et al., 1998b; Dengel et al., 2002; Dieng-Kuntz & Matta, 2002; Lehner, 2000; Stein & Zwass, 1995]). This leads also to a tendency to consider the benefits of formal, automated knowledge-processing as offered by Expert System and related Knowledge Technology approaches, such as Case-Based Reasoning Systems (cp. [Liebowitz & Wilcox, 1997; Malhotra, 2001; Watson, 2002]).

Tools and systems

Example tools and systems for the product-centric KM approach include:

- best-practice databases and lessons-learned archives [Heijst et al., 1996; Weber et al., 2001],
- case-bases which preserve older experiences, e.g., in helpdesk applications, project management, sales support, or in industrial design [Althoff et al., 2001; Bergmann & Schaaf, 2003; Friedrich et al., 2002; Roth-Berghofer & Iglezakis, 2000],
- knowledge taxonomies and formal knowledge structures for Semantic Intranet Portals or Community Portals [Gehle, 2001; Maedche et al., 2001; Spyns et al., 2002a], etc.

Typical projects

Adopting the "knowledge as a product" approach means treating knowledge as an entity which can be separated from the people who create and use it. The typical goal is to take documents with explicit knowledge embedded in them — memos, reports, presentations, articles, etc. — and store them in a repository where they can be easily retrieved. Commonly found types of projects representing this approach are for capturing and re-using:

Typical projects

- **External knowledge.** External knowledge repositories range from information delivery "clipping services" (information push channels) that route articles to executives, to advanced competitive or customer intelligence systems using Information Extraction (IE) techniques for detecting specific events (like changes in a company's board) in a large text corpus.
- **Structured internal knowledge**, e.g. embodied in research reports, product-oriented marketing materials, corporate techniques and methods.
- **Informal internal knowledge,** e.g. discussion databases or "lessons learned" databases.

1.1.3.2 Process-Centric Knowledge Management

Basic assumptions

The "process" approach puts emphasis on ways to promote, motivate, encourage, nurture or guide the process of knowing, and abolishes the idea of trying to capture and distribute knowledge.

This view mainly understands KM as a social communication process, which can be improved by collaboration and cooperation support tools. In this approach, knowledge is closely tied to the person who developed it and is shared mainly through person-to-person contacts. The main purpose of Information and Communication Technology in this case is to help people *communicate* knowledge (not *store* it), to coordinate their work and support collaboration.

Tools and systems

Example tools and systems for supporting process-centric KM include all kinds of Computer-Supported Collaborative Work (CSCW, [Borghoff & Schlichter, 2000, Schwabe et al., 2001; Eseryel et al., 2002]) tools:

- all kinds of synchronous and asynchronous *communication* technology, e.g., e-mail, electronic chat tools, video-conferencing, electronic bulletin boards, discussion groups and mailing lists, application sharing, etc.
- systems for supporting *coordination* of work, such as workflow management systems, group calendars, web-based project management support, shared electronic workspaces

Reference		
[Kühn & Abecker, 1997]; [Mentzas et al., 2001]; [Mentzas et al., 2002]	*Product-centric KM*	*Process-centric KM*
[Hansen et al., 1999]	*Codification approach*	*Personalisation approach*
[Wenger, 1998]; [Hildreth & Kimble, 2002]; [Vicari et al., 1996]	*Reification (based on Representational knowledge view)*	*Participation (based on Autopoietic knowledge view)*
[Sørensen & Snis, 2000]	*Codification*	*Collaboration*
[Trittmann, 2001]	*Mechanistic KM*	*Organic KM*

Table 2: Two Alternative Kinds of KM Approaches

- systems for the optimized *collaboration*, such as technology for distributed authoring of hypertext documents, group-decision support systems, meeting support technology, collaborative information retrieval, etc.
- systems to foster group awareness [Gräther & Prinz, 2001; Wainer & Braga, 2001] and contextualized knowledge sharing [Agostini et al., 2003]
- tools for finding appropriate communication or collaboration partners, e.g. yellow page and skill management systems ([Probst et al., 1999; Benjamins et al., 2002]), as well as sophisticated expertise finder systems (cp., e.g., [Becerra-Fernandez, 2000; Becerra-Fernandez, 2001; Yimam, 2000; McDonald, 2001; Yimam-Seid & Kobsa, 2003]).

Typical projects

Treating "knowledge as a process" usually considers enabling the development and flourishing of communities as a key solution for knowledge leverage. Firms adopting this approach focus on the creation of *Communities of Interest* or *Communities of Practice* (self-organised groups which 'naturally' communicate with one another because they share common work practices, interests, or aims, cp. [Wenger, 1998]), to address knowledge generation and sharing. The emphasis in this case is on providing access to knowledge or facilitating its transfer among individuals. Such projects are heavily depending on the quality of respective management and organization measures to create trust between group members, to facilitate face-to-face experience exchange, to cultivate a good mood within a community, to install appropriate organizational roles for facilitating the community work, etc. (see, e.g., [Wenger, 1998; McDermott, 2000]). Software support is mainly about Community Web Portals.

1.1.3.3 Conclusions

Product versus process view can be found in manifold settings

The dichotomy between product- and process-centred approaches has become evident in various real-world KM initiatives and has been discovered, re-discovered and analysed from different perspectives in the scientific and the management literature many times. A short overview of references poniting out essentially the same difference is given in (cp. Table 2).

Major differences between the two approaches

In (Mentzas et al., 2002), we summarized the general differences between product approach and process approach to KM as shown in Table 3.

	Knowledge as a "Product"	**Knowledge as a "Process"**
View	Knowledge can be represented as a thing that can be located and manipulated as an independent object. Emphasis on capturing, distributing and measuring knowledge.	It is only feasible to promote, motivate, encourage, nurture or guide the process of knowing; the idea of trying to capture and distribute knowledge seems senseless.
Focus	Products and artefacts containing / representing knowledge; usually, this means managing documents & data, their creation, storage, and reuse in computer-based repositories.	KM as a social communication process, which can be improved with collaboration and co-operation support tools.
Strategy	Exploit organised, standardised and re-useable knowledge.	Empower / channel individual and team expertise and skills.
Focus of KM	Connect people with re-usable codified knowledge.	Facilitate conversations to exchange knowledge.
Focus of HR	Train in groups. Reward for using and contributing to data-, document, and knowledge bases	Train by apprenticeship. Reward for sharing knowledge with others.
Focus of IT	Heavy emphasis on IT – mainly document management systems.	Moderate emphasis on IT – mainly on network management systems.
Technologies mainly used	Document repositories, information retrieval, Knowledge DB systems, knowledge maps.	Discussion groups, net conferencing, real-time messaging, push technology.

Table 3: Major Characteristics of Product Approach and Process Approach

It is obvious that really holistic, effective KM endeavours should aim at treating both product-centric and process-centric aspects in an equal, in the optimal way, synergetic, manner.

1.1.4 Current Approaches to KM Software Support

Simple technology can help KM much

As one can see from many reports about KM practices, tools, and success stories (cp. [Davenport et al., 1996; Bullinger et al., 1997; Elst & Abecker, 2003]) Knowledge Management can often be successfully realized using conventional, "simple" technologies.

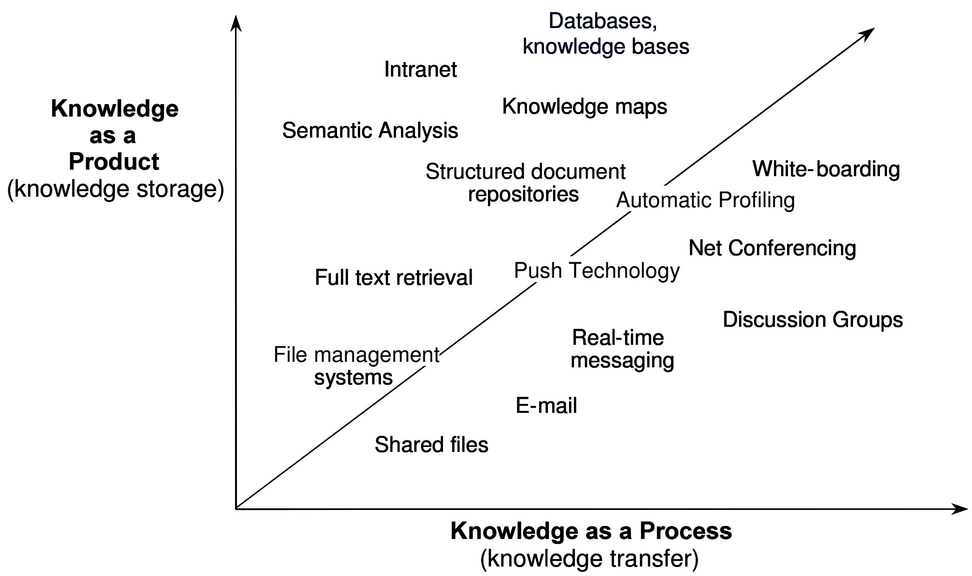


Figure 2: Software Support for Product-Centric and Process-Centric KM

Also advanced KM tools were not especially developed to reflect KM specifics

The advent of Internet and Intranet technologies was one most important *enabler* to start the KM boom, because it allowed new kinds and scales of electronic communication and wide-area collaboration.[1] The deployment of powerful new technologies for Information Retrieval, Text Analysis and Text Classification was another *facilitator* since it made possible highly effective handling of explicit knowledge in Internet sources, in corporate archives, and in so-called "knowledge databases" for, e.g., lessons learned.[2] However, these were technologies neither

---

[1] This was reflected by the commercial success of such tools as, e.g., Lotus Notes.

[2] Typical commercial tool suites in this category were, e.g., Autonomy or Verity.

especially developed for KM purposes nor taking particularly into account the specialties of knowledge as a concept and Knowledge Management as a management discipline.

Figure 2 lists many typical state-of-the art technologies often mentioned and used to support for one or the other KM approach. Though some of those technologies might be innovative and not yet in a widespread use in industrial practice, nevertheless none of those technologies has been developed especially for KM support, nor does any of it aim at an integrated treatment of product-centric and process-centric KM ideas.

Centralized, unifying KM system architecture

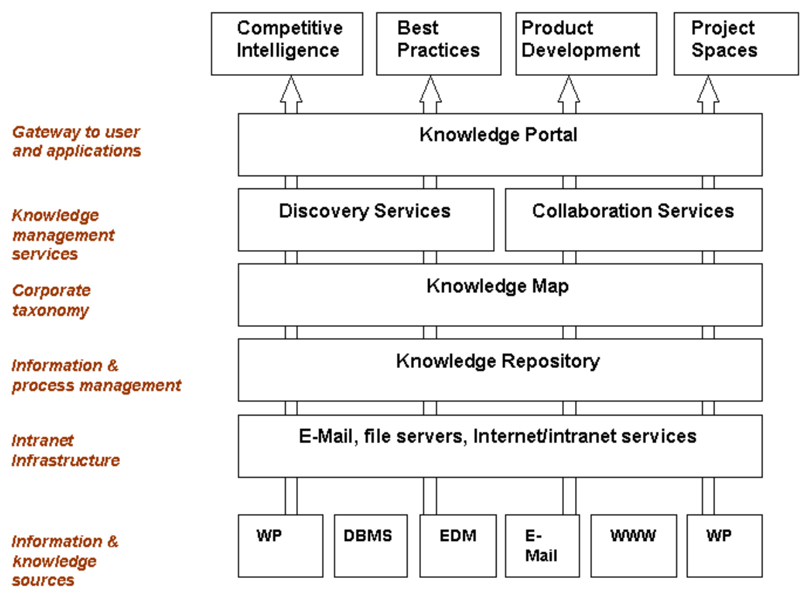


Figure 3: An Integrated Architecture for KM Support

A slight advancement has been achieved with the integrative architectures typically exhibited by big, commercial KM suites which can often be mapped (to the most extent) onto the ideal architecture shown in Figure 3.[3] This architecture makes already some remarkable basic decisions:

[3] The figure has been adapetd from the OVUM architecture [OVUM, 1998] and can be found in several similar reincarnations in manifold publications, see, for instance [Gronau & Laskowski, 2003] or [Paulzen & Haas, 2002]. It was also an input for Maier's excellent analysis of the KM tool landscape and his integrative concept for centralized KM systems [Maier, 2004].

Interesting features of integrated architecture

- It builds – as a repository system – upon manifold different sources, and thus exploits the value of *already existing information sources*.
- It creates a *unifying view* via the so-called knowledge map, or corporate taxonomy, in order to provide a content-oriented integration of different sources.
- It provides *both collaboration and discovery services*, thus addressing to some extent both the process-view and the product-view on KM.
- It supports – through a knowledge portal as the integrated interface – directly a number of *predefined knowledge management processes,* such as Competitive Intelligence, Best Practice gathering, etc.

Our goal in this thesis will be to draw upon those ideas, but also adress some identified weaknesses:

Weaknesses of integrated architecture

- It is not clear how the unifying view through a corporate knowledge map shall be achieved practically, i.e., what is a corporate taxonomy?
- It is not clear whether such a unifying layer that provides the "glue" for integrating existing information sources, couldn't (and shouldn't) also provide more and richer functionalities than just a taxonomy for manual browsing and querying.
- Although there are both discovery services for the product-view on knowledge and collaboration services for the process-view, there is no real integration between both.
- Although the architecture is supposed to support a number of knowledge management processes, it cannot be seen how it directly supports operational, arbitrary business processes.

1.1.5 Requirements from Case Studies

In the 1990'ies, my colleague Otto Kühn did some feasibility studies and industry projects for German and international industry in order to find out the application potential for Expert and Organizational Memory (OM) systems. We presented some insights from those case studies in [Kühn & Abecker, 1997]. It clearly turned out that there were some critical success factors to be addressed inevitably when trying to roll out innovative KM technology in practice. We summarized these critical success factors, or core requirements for KM systems, several times, e.g., in [Abecker et al., 1998a]. Let us shortly review these requirements below:

Case study requirements for KM systems

- **Collection and systematic organization of information from various sources:** Knowledge needed in work processes is currently scattered among various sources, such as paper documents, electronic documents, databases, e-mails, CAD drawings, and the heads and private notes of individuals. The primary requirement for an OM is to prevent the loss and enhance the accessibility of all kinds of corporate knowledge by providing a centralized and well-structured information depository.

- **Minimization of up-front knowledge engineering:** Even though the benefits of having an OM are generally recognized, organizations are reluctant to invest time and money into a novel technology the benefits of which will be far-off. Furthermore, prospective users have little or no time to spare for requirements and knowledge acquisition. An OM thus has to exploit readily available information (mostly databases and electronic or paper documents), must provide benefits soon, and be adaptable to newly arising requirements.

- **Exploiting user feedback for maintenance and evolution:** For the same reasons as up-front knowledge engineering, maintenance efforts for an OM have to be kept at a minimum. At the same time, an OM has to deal with incomplete, potentially incorrect, and frequently changing information. Keeping an OM up-to-date and gradually improving its knowledge can only be achieved by collecting feedback from its users, who must be enabled to point out deficiencies and suggest improvements without causing a major disruption of the usual flow of work.

- **Integration into existing work environment:** In order to be accepted by the users, an OM has to tap into the flow of information that is already installed in an organization. At a technical level, this means that the OM has to be directly interfaced with the tools that are currently used to do the work (e.g. word processors, spreadsheets, CAD systems, simulators, Workflow Management Systems).

- **Active presentation of relevant information:** In industrial practice, costly errors are often repeated due to an insufficient flow of information.

This cannot be avoided by a passive information system, since workers are often too busy to look for information or don't even know that pertinent information exists. An OM therefore should actively remind workers of helpful information and be a competent partner for cooperative problem solving.

In Table 4, we show the major barriers for a successful introduction of KM in German companies, as reported in [Bullinger et al., 1997; Bullinger et al., 1998]. Four out of the top 5 most mentioned barriers – as perceived by the top management of large and medium industry companies, correspond well with critical success factors as we discussed them above.

Mentioned barriers for KM introduction correspond well with our case study requirements

1	Lack of time	70,10%
2	Missing awareness	67,70 %
3	Missing knowledge about knowledge needs	39,40 %
	...	
5	Missing knowledge transparency	39,00 %
	...	
12		27,60 %

Table 4: Perceived Barriers for KM in German Industry[4]

Consequently, the solutions we want to develop in this thesis, should respect as much as possible the requirements from practice mentioned above, thus facilitating to address the problems indicated in such studies as the one underlying Table 4.

1.1.6 Knowledge as a Matter of Information Systems

There is a whole bunch of literature containing theoretical knowledge definitions

It is not a big surprise that at some point in time, a Knowledge Management doctoral thesis is expected to define what "knowledge" in the given context should exactly mean. Nevertheless, this is *not* what we should discuss here extensively. There is already a huge amount of papers and theses discussing this question exhaustively, illustrating facets and perspectives from Cognition, Cognitive Psychology, Social Sciences, and Pedagogics [Rehäuser & Krcmar, 1996;

[4] The rightmost column lists the percentage of answers that mentioned the respective KM barrier as a critical problem.

Reinmann-Rothmeier & Mandl, 2000], Economic aspects and aspects of Organizational Theory [Albrecht, 1993; Kleinhans, 1989; Oberschulte, 1994; Willke, 1996], aspects of Organizational Learning, Organizational Psychology, Organizational Intelligence and Organizational Memory [Buckingham Shum, 1997; Buckingham Shum, 1997b; Lehner, 2000; Matsuda, 1993], discussing the topics of implicit and tacit versus explicit knowledge [Polanyi, 1966; Nonaka & Takeuchi, 1995] as well as collective knowledge [Schneider, 1996], representation of knowledge in computer systems [VDI, 1992; Aamodt. & Nygård, 1995; Richter, 1995; Staab, 2002], different kinds of organizational knowledge [Rao & Goldmann-Segall, 1995], and many other publications. Some of the most often mentioned characterizations are the escalation ladder "data – information – knowledge" (see, for instance, [Probst et al., 1999]) and the Semiotic Pyramid (as cited, e.g., in [Wolf et al., 1999]).

VDI list of knowledge characteristics

A notable list of knowledge characteristics which shows the fundamental problems when dealing with knowledge in information systems, was presented by VDI:

- **Action-oriented:** knowledge is created in the active, lively interaction of an individual with its environment.
- **Subjectivity:** knowledge is created individually in the specific environment of the respective individual.
- **Context dependency:** knowledge is created, acquired, and activitated in the context of specific environmental conditions.
- **Social dependency:** knowledge is created in and through social contacts and relationships.
- **Model-oriented:** knowledge can be differentiated in static knowledge (how things in a problem domain are structured), in inference knowledge (dynamic knowledge and possible problem solving steps), and in control knowledge (how inference steps can be employed to come to a problem solution efficiently).
- **Degree of Consciousness:** knowledge (e.g., in the form of tacit knowledge, know-how, skills) is not always conscious when used for problem-solving.

This list of knowledge characteristics may show that knowledge is not an easy subject to treat in computer systems. Nevertheless, if we want to achieve innovative solutions which take serious the term *knowledge* management, they should be reflected somehow in system approaches and architectures. In order to condense this "wish-list" a bit, we summarize it to few essential points as it was presented in [Scheir, 2002]:

Summary of salient, definitional features of knowledge

- Knowledge is **purpose-oriented** and oriented towards problem-solving.
- Knowledge consists of **networked, contextualised information.**
- Knowledge is bound to **internal models** of people.

These topics will represent essential challenges for our system design.

1.1.6.1 Knowledge Profiles

A spectrum of operating points between information and knowledge

These brief considerations may show that it makes no sense to discuss about the "right" way of representing knowledge, or discuss about the question whether some system really stores *knowledge*, or only *information*, as it is often discussed when people start to design KM and KM systems. Rather it makes sense to see the spectrum of possible knowledge representations which capture the properties listed above to more or less extent, and which represent their individual operating points with respect to costs, efficiency, maintainability, etc. This approach has been followed by [Sørli et al., 1999] with their knowledge profiles.

Bipolar parameters for characterizing knowledge versus information representations

They defined a number of bipolar parameters in order to assess the quality of knowledge encoding as the degree to which the knowledge-centric pole of each parameter scale could be reached and realized. For these bipolar scales, the authors call the left pole **knowledge-centric** (with a strong bearing on learning or acting), and the right pole **information-centric** (unrelated to an actor's adaptive behaviour). Then, the following bipolar parameters are identified: see Table 5.

Subjective vs. objective	Knowledge is always interpreted by an actor, involving a perspective, or a frame of reference. Information, on the other hand, can be said to exist independently of actors. To illustrate this, consider an ancient manuscript written in a hitherto undeciphered script. When scientists then decipher the script, the information content of the manuscript remains the same as when it was written, while lost knowledge is recreated, courtesy of an actor interpreting the information.
Fuzzy versus exact	An actor will often have less than perfect information about its environment. Useful knowledge representations should support non-measurable or limited information, as well as acting under uncertainty.
Associative versus fragmentary (mainly influences acting)	Associativity is a key factor in how the human mind achieves effective knowledge activation. A single key-word may open doors to wide areas of long-discussed knowledge. 'Relevance' as a term is less applicable to in-formation than to an actor's purposive interpretation of it.

Goal-driven versus neutral (mainly influences acting)	Representation and activation of knowledge is always driven by some goal, which an actor wants to accomplish. This has a direct influence on both *what* is stored and *how* it is stored.
Active versus passive (mainly influences acting)	A knowledge representation causes problem solving, or other competent behaviour, to happen when the appropriate context occurs. A knowledge representation must support action relative to brief time windows. Information representations are passive in that they do not in themselves cause action.
Dynamic versus static (mainly influences learning)	Knowledge representations get modified through being used. By formulating an answer or an explanation, you may trigger further reflection that adds new knowledge, even while your information remains the same. Using an information representation, e.g. a book, does not alter it.
Changeable versus rigid (mainly influences learning)	Efficient learning exerts an evolution pressure on the represented knowledge, enforcing revision as new knowledge arrives. Merely adding information to already existing information is not an evolutionary process; indeed, this may even *hinder* the process of extracting knowledge because a large amount of non-integrated information becomes unwieldy in practice (information overload).
Adaptive versus planned (mainly influences learning)	In the real world, unforeseen things happen. A good knowledge encoding should be open-ended and general enough to accommodate reasonable responses to changes in the environment.

Table 5: Bipolar Parameters for Knowledge Encoding, according to [Sørli et al., 1999]

Located in the space spanned by these bipolar dimensions, [Sørli et al., 1999] characterize knowledge in the human brain as the "ideal" knowledge representation as shown in Figure 4.

Ideal means here that knowledge in the human brain is represented in a manner which is equally exploitable for acting and adaptable when learning. Of course, such features would also be optimal for knowledge represented in computer systems. However, there are:

Problems for "ideal" knowledge representations in practical computer systems

- *fundamental problems* (How much subjectivity can be achieved in a system which is not a conscious entity living in the real world?);
- *technical problems* (How to technically implement a high level of associative storage, combined with goal-oriented retrieval?); and
- *organizational or economic* problems (System maintenance must be affordable: How to enable a high level of dynamics, changeability and adaptation in an organizational setting which affects working processes, editorial processes, etc – still achieving economic rationality?).

The "ideal" operating point is defined by the characteristics of the human brain

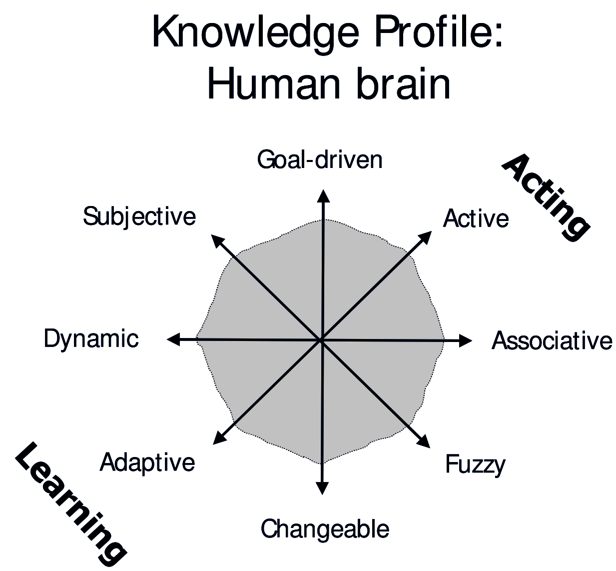


Figure 4: Knowledge Profile of Human Brain, according to [Sørli et al., 1999]

Consequently, today's technical solutions explore compromise solutions as the ones shown in Figure 5.

There, we see two examples (other examples can be found in [Sørli et al., 1999]):

Knowledge profile of a hypertext document

- A *hypertext document* can provide relatively high associative functions, because embedded hyperlinks can directly point to other, related knowledge pieces. As an informal knowledge representation, changeability and adaptivity are realized by manual intervention. Of course, a hypertext document is a completely passive, not goal-oriented way of representing knowledge.

Knowledge profile of program code

- A piece of *program code,* on the other hand, is highly active and goal-oriented, since it can directly lead to active system behaviour, automatic problem (partial) solutions, etc. However, it is even less adaptive and changeable than a hypertext, because adaptation usually means manual, time-consuming re-coding. Associative features are typically also weak, since program code is aimed at solving well-defined tasks in very specific situations, vague associations with "similar" situations are not the typical application profile.

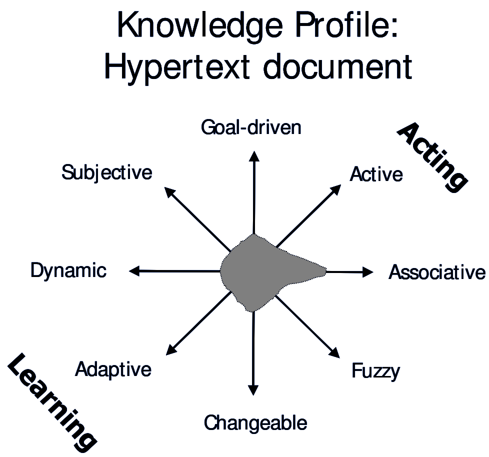


Figure 5: Other Knowledge Profiles, according to [Sørli et al., 1999]

Goal: find practically feasible operating points for knowledge-oriented representations

In this way, we can analyse many more knowledge media, such as e-mail messages, video-clips, expert system code, technical reports as PDF files, etc. A major goal of this thesis is to find out how to come to economically and technically realistic solutions which bring the profile of knowledge in computer systems a bit closer to the profile in the human brain as shown in Figure 4.

1.2 Goals, Approach, and Structure of this Thesis

1.2.1 Goals and Requirements

We gather system requirements from the previous sections

In order to formulate the goals of this thesis, I will summarize the major requirements described in and derived from the previous subsections. The overall rationale is depicted in Figure 6.

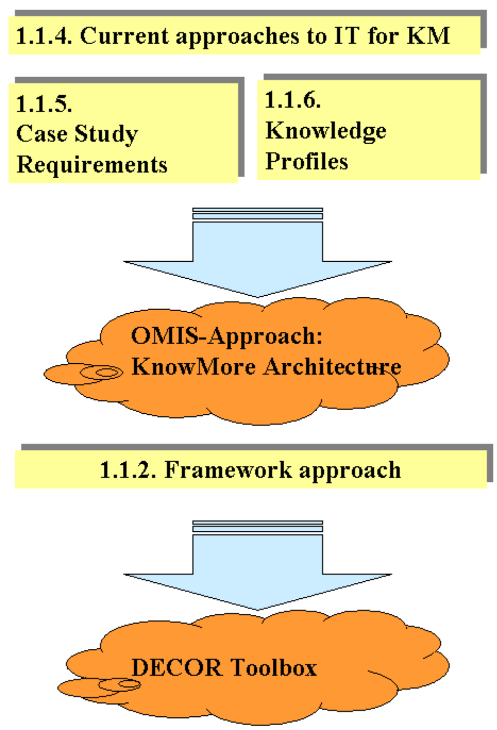


Figure 6: Requirements for this Thesis

Goal 1: advance the state-of-the art in innovative KM software

Starting from current approaches for IT support in Knowledge Management (as sketched in Subsection 1.1.4), we want to keep the basic underlying design decisions and goals (homogeneous access to heterogeneous legacy knowledge sources, support for discovery and collaboration services, portal access for KM processes), but further advance the current state of the art *towards innovative, new software functionalities*.

System approach is driven by knowledge characteristics and by practical requirements

These new software functionalities will mainly be motivated by the typical case study requirements described in Subsection 1.1.5 on one hand, and by the above-mentioned charcteristics of knowledge as a matter of information systems (Subsection 1.1.6) on the other hand.

Organizational Memory Information System (OMIS)

In general, our system approach can be seen in the tradition and as an extension of the concept of Organizational Meory Inforrmation Systems (OMIS). There has been written much about Organizational Memory in general (for a comprehensive overview cf. [Lehner & Maier, 2000; Lehner, 2000]) and about **Organizational Memory Information Systems** in particular (see, for instance, [Walsh & Ungson, 1991; Stein & Zwass, 1995; Watson, 1996]). There have also been some shifts of research focus from early successful OMIS implementations aiming at supporting effective, asynchronous group communication by information systems[5], up to a more comprehensive understanding of an OMIS as a general Knowledge Management software support in an organization. In particular, the use of novel technologies from AI and CSCW has been investigated in manifold forms (cp. [Borghoff & Pareschi, 1998; Abecker et al., 2000a; Dieng-Kuntz & Matta, 2002]).

In order to start the thesis from a clear basis, let us build upon a relatively comprehensive and ambitious definition of an OMIS, motivated by the case study observations summarized above, as well as by holistic KM theory and by the analogy to the human memory. Consolidating, extending, and refining my definitions published in [Abecker & Decker, 1998; Abecker & Decker, 1999; Studer et al., 1999], we can say:

[5] The most important representative of this class of systems was the AnswerGarden tool [Ackerman & Malon, 1990; Ackerman & McDonald, 1996; Ackerman & McDonald, 2000].

Definition: OMIS

An Organizational Memory Information System (OMIS) is

● an intra-organizational computer system that – at the level of a working team, an organizational sub-structure, a Community of Interest, or the whole organization

(SCOPE)

● continuously, pro-actively, and – as much as technically possible and economically reasonable – automatically

(AUTONOMY)

● gathers, structures, stores, and actualizes

(FUNCTIONS: KNOWLEDGE ARCHIVING)

● data, information and knowlegde (from within as well as outside the organization, including already existing information systems)

(CONTENT COMPREHENSIVENESS)

● with different representations, media, content and purpose

(CONTENT HETEROGENEITY)

● and provides its content or derived assistance functionalities to the end user in a pro-active, purposeful, and context-sensitive manner

(FUNCTIONS: KNOWLEDGE EXPLOITATION)

● in order to support general KM processes such as Organization Learning, but in particular also to directly support operational, cooperative, knowledge-intensive business processes and business activities

(KNOWLEDGE-TASK ORIENTATION).

Basic idea: realize an OMIS that realizes the principles of Business-Process Oriented Knowledge Management (BPOKM)

The so-motivated functionalities go into our first specification of an Organizational Memory Information System and its realization in the KnowMore generic OMIS software architecture (or, OMIS reference model), as described in Chapters 2 and 3 of this thesis. The principles realized with this KnowMore architecture are based on the idea of Business-Process Oriented Knowledge Management (BPOKM), since business process management and automation can represent a valuable starting point for software-technological realization of contextuality, pro-activeness, and workplace integration of KM services.

Taking into account that KM projects (as pointed out in Subsection 1.1.2) typically require a holistic understanding that realizes a method-driven project approach and is supported by an integrated landscape of concerted tools, another goal of this thesis is to provide a total solution for Business-Process Oriented Knowledge Management. This total solution was developed with the DECOR ("Delivery of Context-Sensitive Organizational Knowledge") suite of methods and tools which is described in Chapter 4 of this thesis.

Goal 2: provide a total solution (methods and tools) for BPOKM

1.2.2 Research Methodology and Structure of This Thesis

The methodological approach pursued in this work is reflected exactly in the structure of the thesis and can be summarized as follows (see Figure 7):

In Chapter 1 (this, actual chapter), the motivation as well as the scientific and practical background of this work is given. On the basis of a thorough analysis of (i) existing KM approaches and frameworks; (ii) existing IT support for Knowledge Management; and (iii) a well-understood definition of both KM and "knowledge" in general, we derived requirements for Business-Process Oriented Knowledge Management. These can be sketched as follows:

Chapter 1: Introduction

(A) Address the salient features of "knowledge": problem-oriented and purposeful; context-dependent; highly interconnected.

(B) Adress the pragmatic requirements for KM systems: mimimum effort for knowledge engineering; deep integration with existing work practices and tools; proactive, but unintrusive system behaviour; integration of existing information sources.

(C) Provide a total solution, i.e. a framework-embedded, methodology-driven set of tools.

Chapter 2 presents a first answer to challenges (A) and (B): the KnowMore system which is a demonstration prototype already incorporating most of the technical ideas we found relevant for realizing innovative BPOKM systems. In particular, this means a layered system architecture comprising:

Chapter 2: KnowMore Approach

1. An application layer realizing seamless integration of KM support offers into daily operational work, and automatically triggering proactive, context-sensitive activation of KM services.

2. A knowledge brokering layer which provides the machinery for high-precision knowledge retrieval as well as other potentially available, automatic knowledge processing functionalities, such as automatic summarization or automated reasoning over facts.

3. A knowledge description layer that realizes a rich metadata annotation as the integrative platform for allowing homogeneous access to potentially very heterogeneous original knowledge sources, thus providing the basis for the value-adding services of the knowledge brokering layer.

4. The knowledge object layer which consists of all available and accessible sources of explicit knowledge (thus exhibiting a high level of heterogeneity with respect to, e.g., media, storage, conceptual structures, etc.) which could be relevant and useful for KM services.

This layered architecture for Business-Process Oriented KM Systems providing proactive, context-sensitive KM services, is the core of the scientific contribution of this thesis. Since the usefulness, as well as the complexity, of the architecture, mainly stems from the play-together of the several layers and not so much from their exact, detailed implementation and functionality, we followed a two-step approach for presenting it in this thesis (which also reflects the methodological approach of the work undertaken):

- Chapter 2 first presents the architecture as a whole, a bit superficially in the details, but with more emphasis on the overall structures, the rationale and motivations, the interrelationships and interdependencies. To this end, we show the KnowMore system “in action”, i.e.,
  - first, a rough overview of the levels of the architecture;
  - then, two sample applications to illustrate the benefits of such an approach;
  - last, a sketch of the implementation of this first prototype.

Chapter 3: Reference Model

- Chapter 3 then abstracts away from the concrete implementation and discusses each of the four system layers with much more details, thus ending up with a comprehensive, elaborated frame of reference which (i) points out the necessary overall structures for building BPOKM systems, (ii) gives many details to understand potential realizations of specific layers. In this way we describe the design space for BPOKM systems which could lead to concrete instantiations of the reference model that might be completely different from KnowMore.

Chapter 4: DECOR Methods and Tools

Referring to item (C) above – the request for a total solution – in the above list of overall requirements and goals, Chapter 4 then gives an overview of our results achieved in the DECOR project. DECOR's aims were threefold:

- On one hand, consolidating the KnowMore ideas by building a KnowMore-like system upon commercial tools, or, at least, easy-to-use, "close-to-commercial" tools thus paving the way of BPOKM approaches into practice.
- Then, complementing and embedding the overall approach by a (tool-support) method in order to have a structured, proven way of working when introducing BPOKM systems.
- Last, but not least, realizing a couple of case studies in order to give a proof-of-concept for real-world applications, demonstrate the benefits of tools and method-driven BPOKM projects, and better understand strengths and weaknesses of the approach as it is, when applied in a real environment.

These items are reflected in Chapter 4 in respective sections about the (i) overall project; (ii) the modelling method and tools (for process analysis, knowledge-oriented process re-engineering, and knowledge-oriented workflow modelling); (iii) the process-oriented archive system (realizing the knowledge object plus knowledge description layer); (iv) the demonstration workflow engine closely coupled to the archive system in order to realize context-sensitive, proactive knowledge services; and (v) two of the DECOR case studies in a very brief description. We consider the DECOR set of tools and methods the second major contribution of this thesis which shows that Business-Process Oriented Knowledge Management can find its way into practice.

Chapter 5: Research Ideas

Besides the straightforward thread of work described so far, which led from first case studies and requirements, through the KnowMore conceptual work, the KnowMore prototype, the detailed description of the BPOKM reference model, finally to the DECOR set of methods and tools, there emerged several interesting ideas for further research during the couple of years that I spent with BPOKM support. Since it was not appropriate to deviate from the straight way sketched above, some of the promising ideas resulted in other research projects or Ph.D. work pursued by colleagues. We consider these derived research ideas as another major contribution of the author's work to the KM research community – which would not have been made without its grounding in the BPOKM reference model and our application experience with this approach.

Hence we include Chapter 5 in this thesis which contains the most important ideas and approaches developed in this respect, namely:

- Knowledge Trading – the idea of selling knowlede products and knowledge-based products (like teaching courses, consulting projects, technology reports, or technology workshops) over the Internet, and its derived difficulties and problems.
- Agent-Mediated Knowledge Management (AMKM) as an approach to deal with the manifold challenges of Distributed Organizational Memory Systems by the use of advanced concepts for Intelligent Agent software.
- Weakly-structured workflow as a concept to support much better the reality of knowledge-intensive work than conventional workflow approaches do.

These different areas of future work also exhibit a relatively different status of elaboration at the moment. Some of them have to some extent been tackled in ongoing Ph.D. projects (Weakly-structured workflow, AMKM), others led to running European or national research projects (Knowledge Trading, AMKM) or to the just started creation of a new research community (AMKM), or are still open (BPOKM for E-Government). As already mentioned, I consider them as an important contribution of my dissertational work, but not a central element of this thesis, because they are mainly dealt with in detail by other colleagues, and did not yet yield final results. Consequently I address them relatively shortly, and with different approach and level of detail.

Chapter 6: Conclusions

Of course, this thesis ends with concluding remarks (Chapter 6) which comprise, in particular, a short summary of the thesis and its achieved results, a discussion of these achievements with respect to the requirements and goals presented in Chapter 1, and a short discussion of still open issues, besides the ones discussed in Chapter 5.

Since the whole thesis covers a rather broad area of techniques, tools, methods, and approaches, there is no overall section about related work and state of the art, but the respective remarks are made in the specific Chapters, where necessary and useful.

The argumentation structure of the thesis is illustrated in Figure 7.

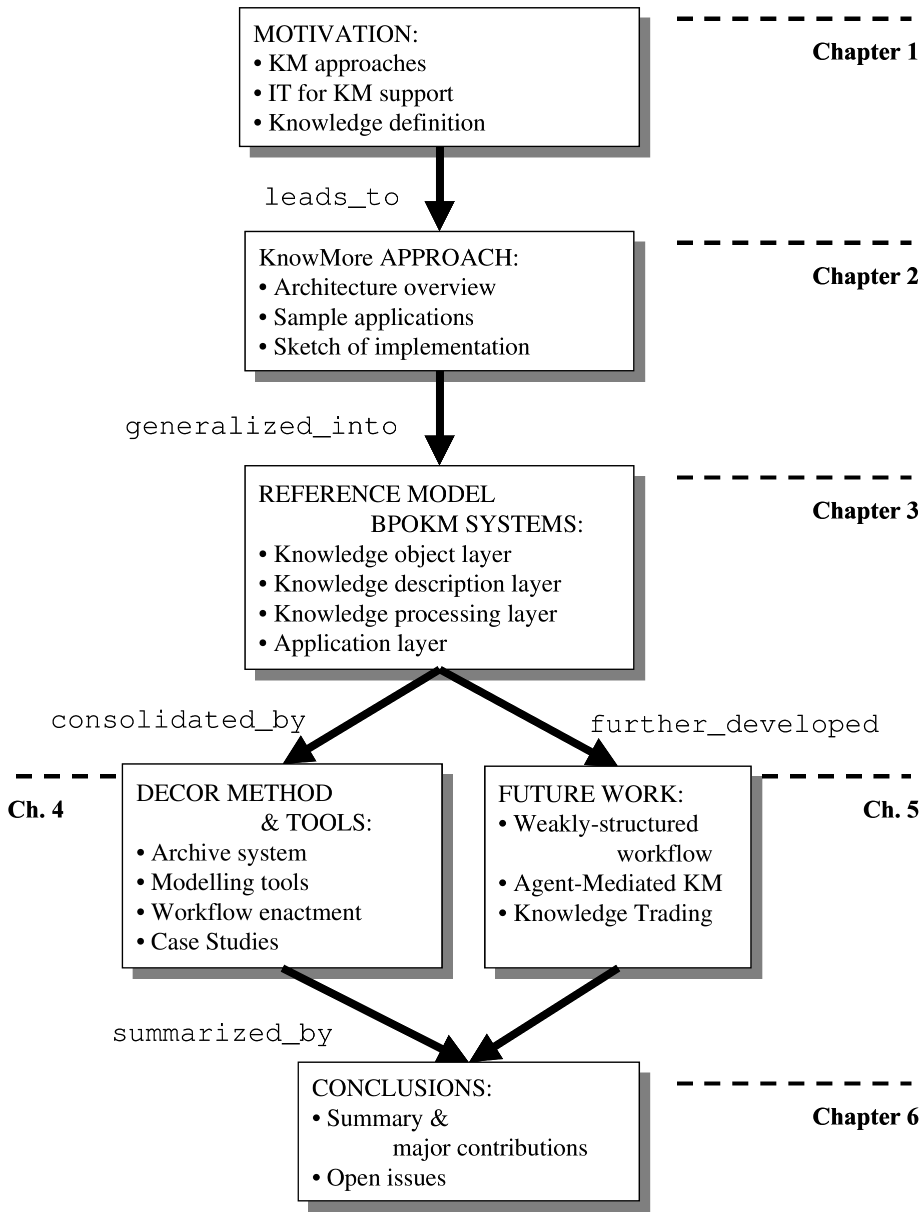


Figure 7: Research Approach and Structure of Thesis

2 The KnowMore Architecture

Knowledge is content in context to produce
an actionable understanding.

Robert Bauer, Xerox Parc

Abstract: This chapter presents the first major contribution of this thesis, the KnowMore four layer architecture for Organizational Memory Information Systems. This reference architecture consists of four functional layers: (1) application layer; (2) knowledge brokering layer; (3) knowledge description layer; and (4) knowledge object layer. Since the added-value of such a layer architecture, as well as its complexity, comes from the synergies between these four layers and not so much from the concrete realization of each separate layer, we organize this chapter as follows:

- First, we briefly sketch (in Section 2.1) the architecture as a whole.
- Then, we illustrate the functionalities achievable by such a system through two worked-out usage examples of the KnowMore prototype - which was the first implementation of (most of) our ideas (Sections 2.2 and 2.3).
- Last, we give an outline about the technical realization of the KnowMore system in order to illustrate how the respective elements of the architecture could be implemented and play together at a techncial level (Section 2.4).

After this review of the overall rationale, the perspective of "KnowMore in action" and an sketch of possible realizations, we are ready for abstracting this into a general framework and reference architecture, i.e. we can then go through the four layers in detail, define them concisely, and discuss possible realizations, in the subsequent Chapter 3.

Preamble: *The KnowMore architecture was mainly developed, documented, and implemented in a first version in the run of the KnowMore (Knowledge Management for Learning Organizations) project at DFKI funded by the German Ministry for Research and Education bmb+f [Abecker et al., 2000a; Abecker et al., 2000b]. The overall approach goes back to my work with Otto Kühn as published in first version as a requirements analysis in [Kühn & Abecker, 1997] and as a first system architecture in [Abecker et al., 1998a; Abecker et al., 1998c]. Nevertheless, the results wouldn't have been the same without the collaboration with many colleagues within DFKI and Prof. Michael M. Richter's working group at the University of Kaiserslautern, and also outside Kaiserslautern, e.g. Stefan Decker at AIFB Karlsruhe. While the focus of my personal work – as presented in this thesis – was mainly on overall architectural issues and the Information Ontology in particular, there were several people with a significant impact on other parts of this work. I hope not to forget anybody important when I say that the use of workflow tools for the application layer was strongly influenced by Knut Hinkelmann, the elaboration of the application layer was presented in the first instance by Ansgar Bernardi, and the whole system (both conceptually and technically) would never have been realized without the restless creativity and extraordinary productivity of Michael Sintek.*

2.1 Overview of the KnowMore Architecture

In this section with give a brief overview of the four layers of our KnowMore architecture for Organizational Memory Information Systems as shown in Figure 8. We list the four layers with their major functionality, the rationale why we think they are required and useful in such an architecture, and their synergetic interoperation in a concrete KM application situation. These four layers will later be explained in much more detail. Also their interoperation will be illustrated extensively in the following two sections at the hand of two worked out examples.

Let us discuss the four layers of our architecture from bottom to top.

Knowledge Object Layer (KOL)

Since the major underlying assumption of this thesis is that knowledge-intensive work can be supported by information systems, we start, of course, from a storage layer holding and giving access to information and explicit knowledge (that is again, stored as information). Most knowledge-intensive work which is subject to improvement by information systems, is essentially *information processing* work, i.e., typically, white collar work in offices.[6] Hence there will occur at least two types of information to be dealt with in our architecture:

- Operational information: information and documents to be handled in the business process under consideration anyway, as input and output of tasks, even before any KM support. Examples are forms to be filled, reports to be created, data to be taken into consideration, documentation of results and drawn decisions, etc.
- Support information: information and documents not yet considered or *systematically* treated in the business process *before* designing KM support, which can be used as additional input to perform tasks more

[6] This is not necessarily the case for all knowlede-intensive activities to be supported by KM initiatives. We could, for instance, also imagine a high knowledge orientation when observing field workers diagnosing, repairing, and maintaining machines in industrial environments (cp., e.g., [Bernardi et al, 1998]). Then, we don't have an office environment or a typical white-collar job. Neverteless, the argument above holds true that the activity fundamentally can be characterized as an information processing activity taking as input and background knowledge general knowledge about the application engineering domain, data and information about the concrete machine installation and its prior maintenance history, technical drawings, technical manuals, etc. So, we have at least virtually an office activity processing paper, data, and other media, even if there is no desk in the literal sense.

effectively, with better quality, more consistent with company regulations, etc. They are typically added to the business activities through the installation of a KM system which tries to systematically collect, evolve, or distribute such information, e.g., best practice documents, lessons learned, etc.[7]

Taking into account that we will define a domain and application independent reference architecture which shall be applicable in manifold, complex knowledge-intensive applications, it becomes obvious that the Knowledge Object Layer must be able to hold manifold different kinds of information objects, represented in manifold formats and media. It becomes also clear that the concrete kinds of formats and media cannot be defined application-independently in advance, and that it must even be extensible at runtime when further developing and extending a running KM system.

Hence the KOL should not be designed as one fixed, monolithic information system or database, but instead consist of a number of information systems, most of them will probably already exist as legacy systems before introduction of the KM system.

Knowledge Description Layer (KDL)

Given this fact, that the KOL of our architecture is rather a virtual system element, comprising a – potentially huge – number of legacy and newly created information systems, it becomes obvious that we need some technical provisions to allow our retrieval and processing mechanisms to access "the" knowledge sources in a somehow homogeneous and uniform maner, even if they are stored in many different systems and representing different kinds of knowledge, different representations, complying with different conceptual structures, etc. To allow such a uniform access to heterogeneous sources, is the purpose of the Knowledge Description Layer (KDL).

---

[7] In practice, when applying such methods as proposed in this thesis, it might even become difficult, for really knowledge-intensive activities, to make the distinction between operation and support information, since the two tend to mix up. Nevertheless, it is useful to make the distinction at least in the methodological survey, to be aware of it in the analysis and design phase of system introduction. Mixing together these two levels is also sometimes a source of confusion when people, especially commercial tool providers, talk about Knowledge Management systems, and it seems that what they provide is "nothing new or specific", just "ordinary *information* management instead of *knowledge* management". In our opinion, it makes sense to make the distinction for methodological purposes, but in a running solution, one needs the two approaches anyway: a streamlined, efficient information logistics for operational information, plus additional specific KM-oriented extensions. We believe that the reference architecture in this thesis supports both types of support equally.

An OMIS architecture for BPOKM

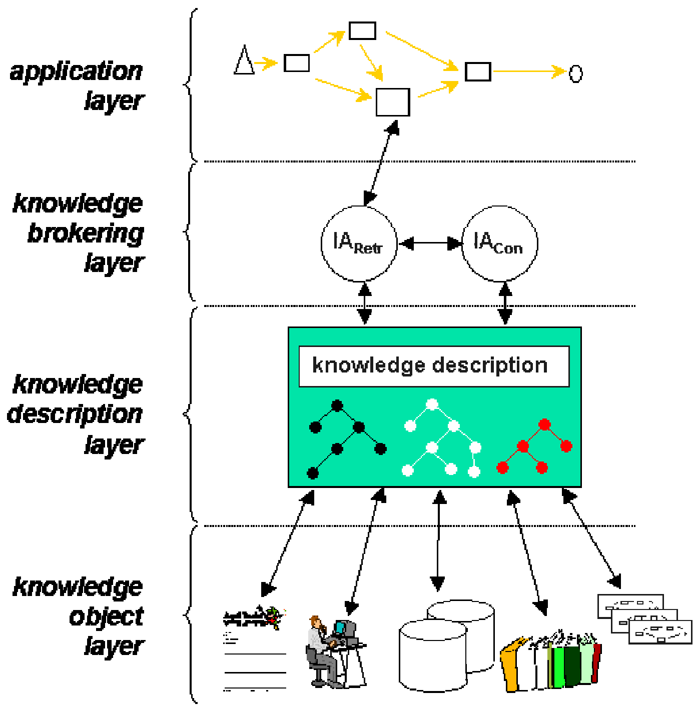


Figure 8: Overview of the KnowMore Reference Architecture

KDL is based on rich metadata

Technically, the KDL primarily consists of a number of metadata frames, each of them representing one concrete element belonging to or creatable from the KOL[8]. Since we assume that "knowledge" is a complex, difficult to explicate and describe, volatile, action-oriented, and highly context-dependent thing which can hardly by caught in an explicit, information-based form, the KDL is a central part of our overall approach with an utmost importance for the question how to "create knowledge out of information". The main assumption underlying our approach is that (maybe, even primitive forms of) manifold kinds of information can obtain the "knowledge status" (at least for the end user working with the system) provided that the employed metadata are chosen in a way that they allow (at least for the end user, even better with a high degree of automation):

[8] "Belonging to" would mean the usual case that a text or multimedia document stored in some archive system is described by a metadata frame. "Creatable from" indicates that it is also possible that some knowledge object might be characterised at the KDL in an abstract manner – by the essential properties which describe its relevance for the given retrieval task – but will just be created at query time when it is required, e.g., by evaluating a complex database query (which might yield different results at different points of time), by evaluating a query against an Internet search engine, etc.

Requirements for knowledge object metadata

- to easily assess the relevance for a given task in an actual work context *(tackles the knowledge feature of contextuality and action-orientation)*[9]
- to re-contextualize a knowledge fragment in the given situation, if it was de-contextualized significantly for storage and automated handling *(tackles the feature of contextuality, situatedness and person-boundedness)*
- to represent and exploit networks, interrelationships and interdependencies between different knowledge objects, also in manifold forms and formats *(tackles the networked character of knowledge)*
- to find and assess the task-specific relevance also of very *informal* knowledge representations – such as personal notes, e-mails, metaphorical war stories, technical drawings, or video clips (*reflects economic requirements*[10]*, eases end user perception, takes better into account the complex character of knowledge*)

Ontology-based metdata in the KnowMore KDL

In order to offer maximum expressiveness for formulating and to allow for maximum automatization of processing metadata, we propose to build the KnowMore KDL upon two basic decisions:

1. Metadata schema and metadata values shall be fully *ontology-based*. I.e., the whole metadata approach is realized within the frame of a formal, logics-based representation and reasoning paradigm which is designed towards information exchange, interoperability, and reusability of models. This shall enable:
   - a high-degree of automated processing,
   - a well-founded semantics of all formalisms and functions,
   - a high potential for interoperability between our system and others,
   - the integration of machine-processable background knowledge about structures and relations in the domain of interest which may support Information Retrieval, processing, or integration,
   - easy extensions by new metadata attributes, domain models, or system functionalities, and
   - the re-usability of techniques developed for Knowledge-Based Systems (expert systems) and logics-based Information Retrieval.

---

[9] The several features of knowledge referred to here, can be seen a summary of the various considerations mentioned in Chapter 1.

[10] In particular, regarding mimization of upfront knowledge engineering and system maintenance costs, as well as deep, unintrusive integration with everyday work.

2. The metadata schema is completely ontologically specified by the following modular approach: The kinds of possible knowledge objects (e.g., lessons learned as semi-structured text, technical drawings, tutorial presentations as powerpoint slide shows, etc.) together with the metadata attributes required for each knowledge object type, are defined in the top-level Information Ontology. The Information Ontology in turn points to two other ontologies describing the ranges for certain metadata attributes (cp. Figure 9):

Metadata are expressed in terms of three basic ontologies

   - The **Enterprise Ontology** shall describe static and dynamic structures (i.e., business processes, tasks) in the organisation considered. It provides values for metadata attributes which shall allow to assess the context-specific relevance of a stored piece of knowledge, e.g., by giving the *creation context* (to which department is the author belonging, in which situation was a lesson learned acquired?) and / or the *potential usage context* (for which task in what kind of business process and which enacting role is a certain regulation appropriate?).
   - The **Domain Ontology(ies)** shall describe *content* of knowledge objects, i.e., what they are about. In a pharmazeutical application, this might be a model of chemical compounds, drugs, diseases, and remedies, in a mechanical engineering domain it might describe parts of an engine and their functional interdependencies.

Following this approach, we can imagine each potential knowledge object as represented by a placeholder in the form of a set of linked instances of these above ontologies. One of these instances is a member of a content type concept, standing, e.g., for a technical report, a lesson learned, a knowledgeable colleague, a training course, etc. This root object has manifold attributes which shall comprehensively describe whatever facet of the knowledge object might be interesting for finding and using it. These attributes may have as their values other ontology instances, representing, e.g., some statement about the subject a technical report is about. Further, we can insert at the metadata level additional attributes talking really *about* knowledge objects, e.g., their quality – as derivable from the author, or their reliability and expected usefulness – as derivable from prior uses of this piece of knowledge.

Lastly, it should be noted that the introduction of a separate, declarative Knowledge Description Level even allows to introduce new, *virtual* knowledge

A separate KDL allows virtual knowledge objects

objects which do not directly represent exactly one physically existing knowledge object, but rather describe the properties of an object that can be created on demand at runtime, e.g.,

- as a composite object gluing together several existing bits and pieces (e.g., agenda, invitation, personal notes, and official minutes of a meeting; or, a legal regulation plus official commentaries and explanatory texts that shall help tp understand and interpret the law correctly);
- as a paragraph of a longer, more comprehensive document, which is extracted when the respective topic sought-after; or
- as a result of a query to a database or expert system, or an Internet search engine which is just interpreted when asked for.

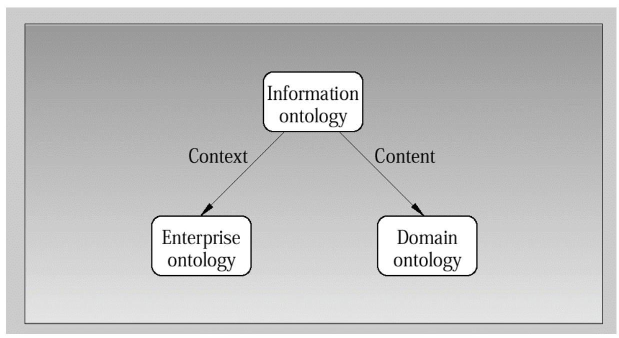


Figure 9: Relationship between Information, Enterprise, and Domain Ontology

Knowledge Brokering Layer KBL

While the Knowledge Description Layer represents a "passive" data structure, all knowledge-processing functionalities required for answering the end user's requests (coming rom the Application Layer; see below) or for realizing additional value-adding services (e.g., for automatic or semi-automatic updating or re-organizing the OMIS), are located at the Knowledge Brokering Level KBL.

This means, all support requests activated by a context-trigger from the Application Layer will be evaluated by knowledge processing services residing at the KBL. To this end, the functionalities of the KBL combine (a) knowledge about the processing of certain support requests coming from the Application Layer, with (b) knowledge about how to access the Knowledge Description Layer and how to further process the results of a KDL query.

In the easiest case, a simple Information Retrieval agent would do nothing more than relaying a given query to the KDL and forwarding the results to the rendering mechanisms at the Application Layer thus presenting them to the user. More advanced levels of functionality could include:

Specialist knowledge in the KBL

- highly-specific **search knowledge** for answering complex requests (e.g., in the sense of Cooperative Information Agents CIA [Klusch et al., 2003; Klusch et al., 2002; ...]); e.g. , it might be an Information's Agent knowledge how to reformulate a query when no answers are delivered (cp. [Stojanovic, 2003b]); or how to assess relevancy of documents by computing the similarity between query context and document-creation context; or how to break down a complex query into several complementary simpler ones which can be combined to a query plan – when, e.g., the result of one partial query says whom to ask for answering another partial query, etc.
- partial **problem-solving knowledge** for processing query results in a way to provide computed answers or answer suggestions. If we allow to arbitrary problem-solving modules in the KBL, our concept can fully subsume the area of Decision-Support Systems (DSS, [Power, 2002]) or Electronic Performance Support Systems (EPSS, [Cole et al., 1997; Brown, 1996]). Respective functionalities would, for example, include data integration and aggregation, as well as Business Intelligence functions.
- handling of **derived information demands**. I.e., it might make sense in certain situations (an example will be given later in Section 2.2) to execute new queries, depending on the results of prior queries, to provide the user with a staged information supply. For instance, if an automatically executable business rule would be able to compute a suggested value for a decision variable, it could make sense to offer to the user not only this suggested value, but also more information about the consequences and details of this decision.
- Last, but not least, we can imagine at the Knowledge Brokering Level **value-adding maintenance services** which continuously try to improve the structure of the organizational knowledge base and the efficiency of the OMIS knowledge services. For instance, at this level, performance and feedback data about efficiency of retrieval algorithms could be gathered and analysed in order to improve indexing structures, to delete useless knowledge objects, or to realize functionalities of Collaborative Information Retrieval (as usual in Recommender Systems).

Application Layer AL

The last layer, but one of the central elements to realize the unique features of our approach, is the Application Layer (AL) of the system. This layer shall build upon and extend the functionalities of a Work Management system[11], i.e. in the most typical realization a Workflow Management system. A Work Management system supports an end user in performing a task which is (typically) solved in a collaborative manner in an organizational context. To this end, the Work Management system *may* provide functionalities such as:

Work management functionalities in the Application Layer

- delegating / dispatching sub-tasks to appropriate people or organizational roles and maintaining or controlling the logical / temporal flow of work items;
- helping people to organize their individual tasks;
- provide the end user with data, information, and documents required for enacting specific tasks and activities;
- starting software tools for the user to be employed for enacting their specific tasks and activities.

A Work Management system (WMS) hosts the Application Layer

Other possible realizations of a Work Management system (besides typical workflow products) may provide stronger support because of a deeper model of the tasks to be supported[12], or they may provide weaker support because in the context of knowledge-intensive activities, it can happen that a deep process and task model is be difficult to get in advance of the process enactment such that computer support might degrade rather to a planning and collaboration support for individual activities.[13] Independent from the concrete realization of the Work Management system hosting the Application Layer of our approach, we have to assume the following prerequisites as given:

1. An *explicit* model of activities to be performed by the user, somewhere represented in the system and accessible.

[11] The notion of a Work Management System is unusual. Nevertheless, we use the term here in order to indicate that the *Workflow Management System* that we discuss in the remainder of this thesis, is only one possible instantiation. However, one could imagine also other realizations (see below).

[12] For instance, task-specific Expert Systems could be mentioned here ([Förtsch, 1996; Förtsch, 1998] describes a Design Support system for Mechanical Engineers with an expressive context modelling on the basis of a thorough analysis of the constructing task in mechanical engineering), as well as Knowledge-Based Performance Support Systems ([Reimer et al., 1998; Reimer et al., 2000]). We also presented prototypes of such Work Management systems on top of Expert System technology in [Kühn & Abecker, 1997; Tschaitschian et al., 1997]

[13] We discuss this phenomenon in some more detail in Section 5.3.

2. A *history* of each running business process or activity instance currently being supported by the Work Management system.
3. An *interface to the end user* where open tasks, maybe tools to be used, documents to be processed or information to be consumed, are offered to the user.

Now the KnowMore Application Layer extends these elements and its functionalities as follows:

AL-extensions of the WMS

- Activity models are extended *at modelling time* in such a way that they contain not only, say, task name, starting conditions, software or hardware resources required, data perspective for tools, etc, but also **task-specific information needs**, maybe parameterized by runtime-dependent context variables.
- The *runtime history* is accessed in such a way that – when a specific task is started and the respective task-specific information need is interpreted by the retrieval algorithms of the KBL as a query against the knowledge stored in the KOL and described in the KDL – the retrieval algorithms can exploit the actual values of context variables in order to realize a **context-specific information search**.
- The *end user interface* is extended such that the so-found potentially useful, context-specific information is offered to the user in an unobtrusive manner as a **pro-active support offer**.

KnowMore realizes proactive support using a dynamic task context

Altogether, the so-realized system realizes a proactive information support on the basis of a dynamic task-context. This, together with the extensive ontology-based information source modeling, is the major innovative feature of our approach. Now, the following two Sections shall give an impression how this looks like in practical use.

2.2 The KnowMore Purchasing Application

2.2.1 Process Analysis

As the first worked out example for the use of the KnowMore prototype which implemented the ideas sketched above, we consider the business process of purchasing goods in a company. Indeed, for realizing this prototype at DFKI, we settled upon an already existing cooperation with the DFKI and the University of Kaiserslautern purchasing departments, in order to acquire the necessary process knowledge for the implementation of a real-world suited system (cp. [Abecker et al., 2000c; Baumann et al., 1997]). Figure 10 shows the respective business process, graphically depicted with the ADONIS business process modeling tool (cp. [Karagiannis et al., 1996; Junginger et al., 2000]).

Overview of purchasing process

We summarize the main steps of this business process :

- Some employee, say, a researcher in the KM department, starts the process by filling a demand specification form, stating, e.g., that he needs a high-end PC with good graphics processing abilities.
- In the next steps, the person responsible for the department budget, typically the department manager, checks whether the (roughly estimated) costs for buying such a PC can be covered by the department budget, and hands over the demand specification to the department director. The director supports or rejects the demand.
- In the case of support, the process is split into two alternatives: if the good to be purchased, is hardware or software, the next step is performed by the company's IT infrastructure group. In the case of some other good (like a new desk, an office chair, or a coffee machine), the process is handed over to the general purchasing office of the company. In both cases, the next step is to prepare a detailed demand specification. This means, in the above graphics PC example, that some knowledgeable person in the IT infrastructure group has to translate the rough, maybe underspecified, demand specification of the original end user (like "a high-end PC with good graphics processing abilities") in a concrete product selection: buy what machine from which supplier?

- After this detailed demand specification, the exact price of the good to be bought, is clear. In the case that some financial limit is exceeded, there has to be another approval by the company's purchasing office.
- If there is a final "buy" decision, an order is sent out to the specified supplier, the purchasing database is updated, it has to be waited until the good has been delivered, etc, etc.
- The following steps consider things like paying the bill, assigning an inventory number, installing the good (if it is hardware or software) etc. They are not so important anymore for our KM example.

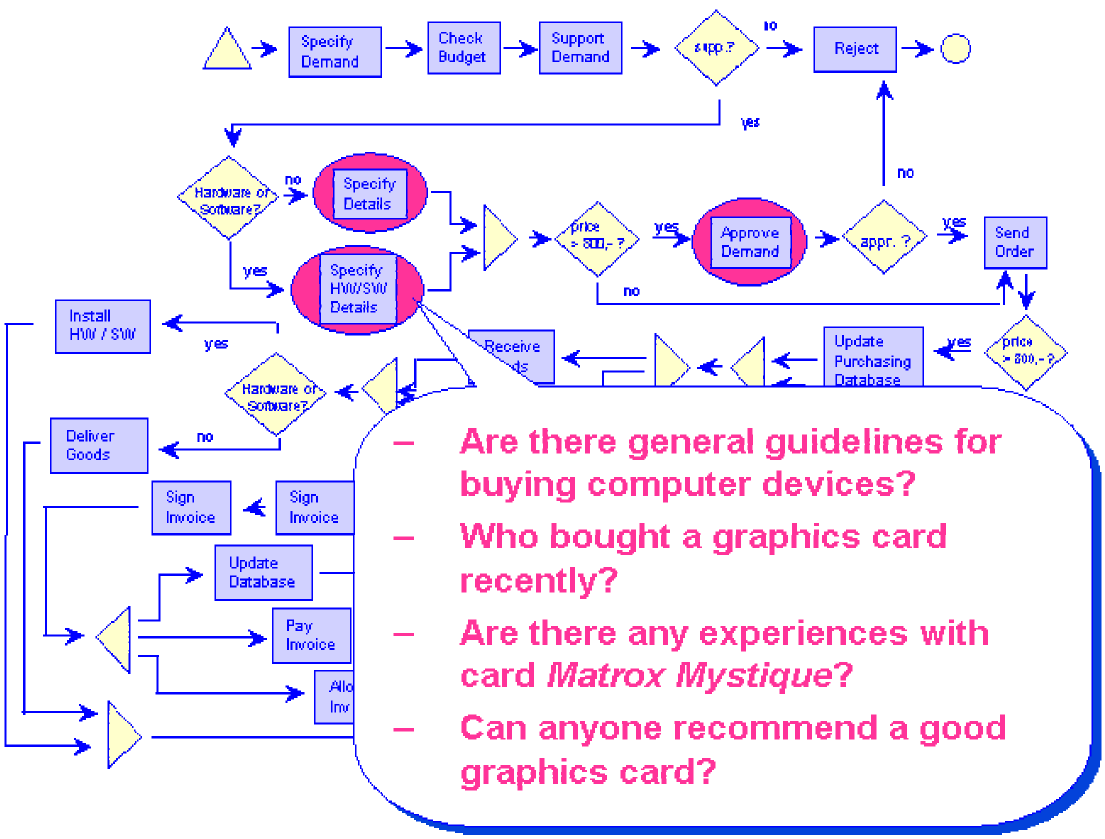


Figure 10: Purchasing Process and Associated Support Potential

Organizational Role or Position	**Most important activities**
Employee of research department X	• Fill in demand specification
Budget manager of department X	• Check budget
Director of department X	• Approve demand • Sign invoice
Employee of purchasing office	• Approve demand • Detailed demand specification for no-IT goods • Send order • Assign inventory number for non-IT goods • Update purchasing database • Receive non-IT goods • Deliver non-IT goods to department X
Senior engineer within IT infrastructure group	• Detailed demand specification for IT goods
Clerk in accounting department	• Pay invoice
Junior engineer in IT infrastructure group	• Assign inventory number for IT goods • Install hardware/software • Update HW/SW database

Table 6: Roles and Activities in the Purchasing Example

Affected organizational roles

Table 6 enumerates the organizational roles involved in our scenario and the respective activities they perform. For the purpose of this thesis, it is not really necessary to discuss and model these roles in detail[14], but we want to show that

[14] Although, of course, a workflow system for KM support must offer a role concept, in particular, since organizational roles are especially important for defining rights and obligations in KM. It is easy to imagine that there might be knowledge of different kinds – specific lessons learned and personal experiences with certain products, suppliers, or employees of suppliers, corporate regulations regarding purchases or negotiations with business partners, information about the financial liquidity of the own company or of a supplier, ... - which should not go outside a specific department or level of hierarchy in the company.

already in such a relatively simple and everyday enacted business process in a small company, the effective collaboration of a number of people is required who exhibit a pretty different status of personal knowledge about the affected issues (department environment and usage situations for the good to be bought, financial data, product knowledge, purchasing process knowledge, knowledge about suppliers) as well as pretty different information needs to effectively perform their respective tasks.

Knowledge-intensive activities in the process

Having a closer look at our purchasing process, together with the potential for KM support in mind, it turns out that, among such simple administrative activities as sending a letter, assigning an inventory number, updating a database etc., there is only a few working steps really requiring expert knowledge and purchasing experience. Some of them are marked in the picture by a dark surrounding circle. We will call them knowledge-intensive activities[15] in the following.

We will focus here on the preparation of the detailed demand specification of the goods to be purchased (which computer model from which manufacturer delivered by which supplier?) based on the possibly rather vague demand description of the employee who initiated the purchasing process („I need some high-end PC with a good graphics card!“).

Potential knowledge support

If an unexperienced employee should accomplish such a detailed demand specification, questions like the ones shown in the center of Figure 10 could arise, the answering of which would be a helpful service of an Organizational Memory Information System. They could address, for instance, the following parts of the organizational knowledge base (cp. also Table 8):

1. <u>Company regulations</u>: is there an obligatory procedure? Are there defined business rules (like, e.g., we have always to buy hardware or flight tickets from one out of a set of three suppliers, because they offer us special prices)?
2. <u>Lessons learned:</u> was there some documented recent experience with a purchasing activity similar to the actual one (like, e.g., did anybody buy the same graphics card that I am currently considering)?

---

[15] The notion of knowledge-intensity has a long tradition in Knowledge-Based Systems (KBS), e.g. when regarding the CommonKADS methodology for engineering KBS [Schreiber et al., 1999]. For this thesis, an intuitive understanding of the concept will be enough for the moment. We will come back to this term, later.

3. Personal concrete experiences: did anybody buy recently a similar good as I want to buy (which could be found out by investigating the purchasing database or the workflow history for the purchasing support system)?
4. Personal specific skills: is there any expert in the company with specific knowledge regarding the good I am currently investigating for purchase (which could be found out inspecting a yellow page or skill management system in the company)?

In order to build a system which would be able to proactively offer answers to questions like those, in a context-sensitive manner, within the running workflow, the following steps have to be performed:

Modelling knowledge support

1. During *business process modeling*:
   - Model the overall **business process** with a conventional BPM / workflow tool.
     - This involves in particular: modelling of tasks, including required resources (tools & input information) and responsible organizational roles, as well as modelling control flow.
   - Identify **knowledge-intensive activities** and associated potential KM support.
     - Typical characteristics of knowledge-intensive activities are (i) the processing of manifold data and information from different sources; (ii) often that a decision has to be drawn; (iii) often that such a decision requires not only a yes/no-answer, but rather requires *constructing* or *designing* a solution; (iv) often trading between conflicting goals; (v) sometimes negotiating with partners and stakeholders in a decision process.
     - For defining KM support requests, identify questions the answer of which could support finding an optimal decision and the anser of which could be found or facilitated by a query to the organizational knowledge base.
   - For each **support request**, create an OM query and attach it to the respective activity.
     - Normally, such a query can only be formulated as a *generic* query the concrete, detailed specification of which can only be given at

runtime by instantiating some variables from the running workflow context.

- Analyse those **context variables** which might influence the concrete formulation of OM queries and which are able to transport context information between process steps in order to formulate OM queries as specific as possible.
  - For instance, in the above purchasing example most queries to the OM would be by far too abstract for yielding useful results, if we would not instantiate at runtime which concrete good should be purchased. It is nonsense to ask for usage experiences regarding *some* hardware, it is probably still useless, to ask for usage experiences with *some* graphics card, but it might lead to interesting results to ask for usage experiences with graphics cards produced by a certain manufacturer, or ask even for a specific, concrete graphics card.
- Create a **workflow model** to enact the modelled business process which also contains the respective context variables necessary to detail OM queries at runtime in a context-sensitive manner.
  - In the example, “normal” **workflow control variables** which are required for dispatching the control flow of activities, are, for example: demand_supported_by_mgt? (boolean), hardware_or_software? (boolean), price_above_800? (boolean), demand_approved? (boolean), ...
  - Regarding **context variables,** we modelled in the example presented in this section, the following variables (see Table 7).

Variable	**Set in activity**	**Used to which purpose**
Quantity (integer)	Initial demand specification	- Price computations
Product_type (ontology concept)	Initial demand specification	- Find general purchasing regulations - Find information about product classes
Product_name (ontology instance)	Detailed demand specification	- Find information about concrete product - Find suppliers

Price (real)	Detailed demand spec.	- Price computations
Supplier (ontology instance)	Detailed demand specification	- Find information about supplier

Table 7: Context Variables in the Purchasing Example

Realizing knowledge support

2. During <u>*workflow enactment*</u>:
   - The system interprets the workflow model step by step and stepwisely instantiates and transports values of context variables.
   - Coming to a knowledge-intensive activity, the system instantiates generic OM queries with context information specific to the actual workflow instance, tries to answer the queries through the OM, and actively delivers (or, offers) the answer to the user.

Before we can discuss the enactment of the so-modelled workflows with our demonstrator system, we have to describe the knowledge sources which were contained in the OMIS knowledge base of the sample application. The most important knowledge sources are listed in Table 8 below.

Type of knowledge source	**Content and usage characteristics**
Business rules (stored in system RULES_DB)	- General rules about purchasing strategies - Applicability depending from product type (context variable: specification or product, respectively) - Can be executed (partially) automatically - Can lead to mandatory or recommended concrete purchasing decisions - Have different levels of binding character
Experiences and lessons learned (stored in system NOTES_ARCHIVE)	- Specific experiences (made inhouse or received from elsewhere, like technical magazines, or partner companies) - Applicability typically depending on concrete value of context variable product or supplier - Usually text document, to be interpreted by the user - Typically high relevance for decision, if applicable

Type of knowledge source	Content and usage characteristics
Supporting documents (stored in system NOTES_ARCHIVE)	- Articles and documentation from technical magazines and journals, or from web sites (e.g, from HEISE newsletter, or c't articles) - Retrieval depending on product class or concrete product instance, or manufacturer (context variables: specification, product) - Different document types applicable in different situations: product comparisons or general technical articles about product classes or functionality of specific technical realizations, in early phases of decision process, specific product tests in later phases
Technical documentation (stored in system PRODUCT_DATABASE)	- Concrete product data, typically found on manufacturer websites
People (described in yellow page system ENTERPRISE_ COMPETENCE_BASE)	- Homepage / yellow page of people who are knowledgeable about (i) the purchasing process, about (ii) some product class (HW/SW, consuming goods, ...), or about (iii) some specific product or supplier - Their expertise might be explicitly mentioned in the yellow page, or it might be derivable from some other information, e.g., because they were the last one to buy such a good

Table 8: Content in the Purchasing Knowledge Base

2.2.2 Modelling Information Needs in KnowMore

Since the task-specific description of support requests and dynamic context-variables is the core concept that enables intelligent assistance for knowledge-intensive activities, we describe in some detail how this is realized in the KnowMore approach. First, let us consider a general schema how knowledge-intensive activities and their associated support requests (also called information needs or support specifications) could be modelled. Such a general schema is given in Table 9 below.

KIT - Context information

describes general attributes of the Knowledge-intensive task, inherited from simple task descriptions

```
Name,                 // a symbol identifying the KIT
execute,              // the application software to be started
input:{variable},     // local task-context of KIT
output:{variable},    // local goals of KIT (decision variables)
```

Support specification

contains a set of information needs which connect between decision variables and calls to information agents

```
Local-variables:{variable},    // local variables used
                               // within the KIT description

infoneeds:{                            // a set of information needs
     (name,                            // a symbol
      description,                     // a comment
      precondition:{constraintobject},   // a set of constraints on any
                                         // of the variables accessible
                                         // inside the KIT.

      agent-spec,    // a string which denotes a generic query to
                    // an information agent

      parameters:{variable},   // parameters to be handed over
                               // to the information agent

      from:{info-source},      // optional: sources to be consulted

      contributes-to:{variable}     // local or output variables to be
                                    // filled by result of query
      )                        // end of information need
            }                  // end of list of information needs

processing:{                   // set of post-processing rules
      if {constraintobject}   // production rules, depending on the results
                             // of queries, as well as variables
      do {action}             // post-processing the results
            }
```

Table 9: Description of Knowledge-Intensive Tasks + Their Support Requests

Technically, a KIT model is a specialization of an ordinary workflow task model, extended by a *support specification* that contains information needs and processing rules, which may refer to the global and local process context. The support specification fills a description frame as shown Table 9. This description frame specifies:

Preconditions of information needs

The **precondition** allows to restrict the evaluation of information needs depending, e.g.:

- On the state of their parameters: only execute if some variables are already non-null. For instance, I can only obtain information about suppliers of a specific product, if I know already which product I want to buy.
- On the state of their parameters: if a specific parameter is already known, skip this information need. For instance, if I have already a final decision for a specific product, I don't have to think about introductory articles about this kind of good.
- On the state of the process: skip if time is critical. If I see, for example, that I need a decision within one hour, I don't have to consult a commercial information service which always takes some days for an answer.

Agent specification of an information need

The **agent-spec** description of the relevant information describes a specific information agent. Such a software agent is responsible for retrieving a specific kind of relevant information, typically from one information source. At runtime, it is invoked and provided with a number of *parameters* taking context information from the actual working situation to the retrieval process. In principle, such information agents could realize complex behaviours, if they possess themselves information-seeking expertise and / or problem-solving expertise.

Source specification of an information need

Sometimes it might be the case that already at process analysis and modelling time we know exactly form which information sources a needed information can be selected. In these cases, the **from** parameter allows to specify this in the information need specification. In principle, determining the information sources which are relevant for a particular information need could also be seen a central objective and significant part of the "intelligence" of the information agent. Hence, in advanced implementations of KnowMore-like systems there might be examples of information agents which are not provided directly with such a from-specification, but rather determine the relevant information sources themselves by computing

info-source = f (parameters,expected-output,callingActivity,processInstance)

depending form the activity's goal and context information. However, if we can identify suitable information sources at process definition time, we should represent this knowledge directly.

Contributes-to describes the goals of an information need

The **contributes-to** field indicates the goal of the particular information need, i.e. that decision variable the filling of which shall be supported by evaluating the given information need. The simple use of this information is to indicate on which places in the application software the results of this information need should be offered to the user. This means concretely in our demonstration examples, at which places in the KnowMore variable editor an information button "I" should occur which indicates that there is some system information available (see below, in the explanation of Figure 11). Further, on the basis of the contributes-to information, the interconnection between different information needs can be deduced and evaluated by the system. For instance, if one information need describes how to find details about suppliers for a given product and another information need produces a suggested value for the product to be bought, it makes sense to evaluate the latter information need before the former one.

Processing rules for output post-processing and information need chaining

The **processing** rules are a number of forward-chaining rules that govern the postprocessing of the retrieved information. At least three cases are possible:

- PRESENTATION: The result of evaluating an information need is presented to the user. In this case, an ordering for relevance-based presentation must be determined before (see below when explaining Figure 12 and Table 11). This can be done using processing rules.
- PROCESSING: In certain cases, however, an information agent might possess some problem-solving knowledge of its own, e.g., specifying further operations for creating added value from retrieval results. A simple example might be that retrieval produces a number of numeric values from a database and postprocessing defines some analyses or aggregations, like computing the sums, the average value, or the median of results.
- INFO NEED TRIGGERING: The result of some information need can also trigger further information retrieval operations, i.e. activate other information needs. For instance, having determined the number of products potentially relevant for a given purchasing decision, we could ask for suppliers of those products, for product data or usage experiences, or for product comparisons for these products.

For formulating preconditions and for describing post-processing rules, we can use **constraintobjects.** These may contain boolean expressions about the values of any variable accessible inside the KIT, or about meta-information which is provided by the system when evaluating an information need. Examples of such potentially useful meta-information is, for instance:

Constraintobjects describe conditions for actions of an information agent

- That some information need produced *no result.* For example, if access to written experience in the product database, the Internet, and the lesson learned system yields no information about a given product, it might make sense to consult an expert by querying the yellow page system.
- How many results were found. For instance, if a huge number of potential suppliers is found for a given product, it might make sense to perform an extensive (which means, time and resource consuming) price comparison, because there is some probability that there are significant price differences between so many vendors.

Last but bot least, an **action** describes what should be done when a processing rules is applied, e.g., sorting of results, calculation of values, the setting of variables, or the activation of derived information needs.

Actions describe effects in post-processing rules

Now we can have a look at Table 10 which shows a sample use of the modelling approach. This example illustrates a knowledge-intensive task model which could be taken from the purchasing example, describing the task of specifying details of a hardware/software purchase, with its associated information needs:

- In the example, the only relevant context variable for the given information needs – described by the **relevant-input** clause – is the product_type, i.e. the kind of product given in the initial demand specification.

  Context variables are information need input

- Further, the clause **expected-output** states that the goal of this task is to find values for the variables product_name (the concrete product to be bought), price (an estimated price for the purchase), and supplier_id (the suggested vendor). This has the effect that those variable occur as editable in the KnowMore variable editor (cf. Figure 11) and that the related decisions may be supported by information offers.

  Decision variables define the point of application for information offers resulting from evaluating information needs

- Then we have an unconditional – i.e. always evaluated – information need called **available-products** – characterizing a simple database access with some background knowledge about class hierarchies in the product

  Information need "available products"

taxonomy – which supports the selection of a concrete product by finding all available concrete products in the existing product database which are instances of the product type or some subclass of the product type written down earlier in the initial demand specification.

```
KIT:
( name:           Specify-product-kit,
  relevant-input: {product-type},
  expected-output:{product-name,price,supplier-id},
  infoneeds:{
   (name:       available-products,
    description:"Products of the wanted type, from database",
    precondition:{},
    agent-spec:  "database-agent select $p"
    parameters:  {product-type},
    from:        {product-database}
    contributes-to:{product-name}
   ),
   (name:        ask-specialist,
    description: "find a specialist for the wanted product type"
    precondition:{product-name==null} // ask only if no idea yet
    agent-spec:  "person-competence-agent",
    parameters:  {product-type},
    from:        {enterprise-competence-base}
    contributes-to:{product-name,supplier-id}
   ),
   (name:        relevant-suppliers,
    precondition:{product-name!=null},
    agent-spec:  "database-agent select($p-type,$p-name)",
    parameters:  {product-type,product-name}
    from:        {list-of-suppliers}
    contributes-to:{price, supplier-id}
   ),
   (name:        previous-experiences,
    precondition:{},
    agent-spec:  "full-text-retrieval keywords $*",
    parameters:  {product-type, supplier-id}
    from:        {notes-archive}
    contributes-to:{product-name, supplier-id}
   )}
  processing:{
    if (price>100) propose previous-experiences
    if (supplier.specialconditions==TRUE) price=0.98*price
  }
)
```

Table 10: Example KIT Description

Information need "ask specialist"

- Next, we have an information need **ask-specialist** which consults the company yellow pages to find a colleague knowledgeable in the considered product_type area who could give valuable hints for both product and supplier selection. Since this is a relatively expensive action (it costs the working time of another employee), it is only activated if neither the user nor another information agent has already found a value for the decision variable product_name such that one needs some stronger support.

- The information need **relevant-suppliers** is again a relatively easy database access which can be evaluated in cases where the names of concrete products to be bought are already known, and where from a list or database of potential vendors those are selected who offer the interesting product or some product in the same class of products.

Information need "relevant suppliers"

- The last information need, **previous-experiences**, might perform a thesaurus and ontology-based search for lessons learned in some experience database, and could deliver interesting notes in the case that there might have been good or bad experiences with some specific product or some supplier providing some type of good.

Information need "previous experiences"

- Finally, we see two simple **processing rules**. One is just checking the expected price of the overall purchase and, in the case that some limit is exceeded, ranks the personal experiences with products or suppliers higher in their relevance (the "propose" macro indicates that the relative importance of some information result is increased for presentation to the user). The other rule just adjusts the estimated price by some discount, provided it can be found out that the specific supplier selected offers some special conditions for our company.

Two simple post-processing rules

Altogether, we see that some of those information needs and evaluations may be very simple, easily implementable system services, like pretty conventional database queries, while others might hide complex information-retrieval algorithms on the basis of ontological background knowledge etc. However, the really important message is that the relatively simple mechanisms for controlling the applicability of information needs, for processing results, and for chaining several information needs depending on the results of prior computations, all in all constitutes a simple to use, yet extremely powerful, apparatus for defining information services. This is the really interesting feature of the KnowMore approach.

2.2.3 Runtime Support With the KnowMore System

Implementation of sample application

In order to demonstrate the benefits of the KnowMore approach without the need to install and maintain a full workflow application plus associated task-specific application software, we built a demonstration prototype as follows:

The KnowMore workflow engine

- A simple workflow engine (KnowMore-WFE) was implemented which was able to enact the usual workflow control primitives as required in the application example (such as sequences of tasks, AND-splits, OR-splits, conditional branching, and loops), and which realized a simple role concept for user management and dispatching tasks to people. Each user was equipped after logging into the system with a simple task-list handler offering the currently open tasks dispatched for the organizational role that this user enacts. When a user decided to work on a specific task, this task was locked for other users, and the KnowMore variable editor (see below) was opened as a mock-up for some arbitrary application software which could be used in a real-life application of a workflow system. After finishing the task, the user committed the results by closing the variable editor, and the workflow interpretation proceeded with the next step.
- The KnowMore-WFE managed the workflow control variables together with the context variables.

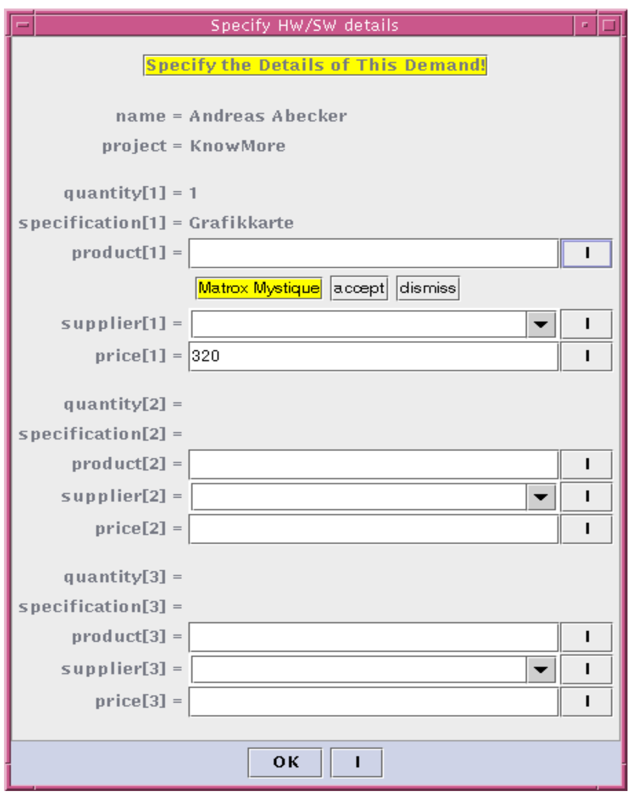


Figure 11: The KnowMore Variable Editor

The KnowMore variable editor as a mock-up application

- We did not care about integrating real-world software applications, but provided a mock-up for all end-user tools to be called in the workflow. Since the essence of enacting workflow tasks is to receive information, process it, and finally come to decisions which lead to the manipulation of decision variables, we mimiced all end user – application interactions with a simple variable editor as shown below in Figure 11: the interface presented to the end user is just a list of variables to be decided on in the run of the current workflow instance. In the case that a specific variable has already been set, it cannot be changed anymore; in the case that a decision about the value of this variable has to be drawn in the actual process step, it is editable in the window.

Some more details about the implementation of the KnowMore prototype can be found below, in Section 2.4, or in the respective technical documentation [Abecker et al., 1998d; Abecker et al., 2000d].

Working with the KnowMore prototype

Now let us assume that each variable mentioned as one row in the variable editor in Figure 11 represents one knowledge-intensive decision to be drawn in the run of the workflow. In the given example, the user is working on the task of preparing the detailed demand specification, i.e. concretely, to set variable values for the variables product and supplier. In the given situation, the name and the project of the requesting end user might have been provided in the initial demand specification ("Andreas Abecker" and "KnowMore"), as well as the initial demand specification ("Grafikkarte").

Information offers

Provided that in the process modelling phase certain variables have been marked as knowledge-intensive decisions and have been equipped with support demands (i.e., OM queries the answers to which might be helpful for the given decision), now an "I" occurs besides these variables in the editor. Following the principle of unobtrusive user interfaces for assistant systems, we do not push an information overload onto the user, but merely offer him some support he might accept, or not.

Computed values

There is one exception to this rule: if there is a support offer which is able to deliver not only support information, but can derive a complete suggested answer for the given decision problem, and if the respective support is marked in its metadata as sufficiently important (for instance, because it represents a mandatory business rule), it might make sense to insert a suggested decision directly into the variable editor.

In the example, this is the case for the decision which product to buy. Assume we have a business rule which states that for the product class of graphics cards, it should always be bought the same product as last time (assuming that the first buying decision was made with reasonable effort and carefulness), except a new product line arose since then. All conditions mentioned above hold true in this case:

- The rule can be fully automated, since the only necessary step is to access the purchase database and find out the last purchasing event involving a graphics card.
- The rule might be marked as a strong recommendation since it saves time, might save money because of special conditions with a certain supplier, reuses the positive experience from last time (provided a negative experience would have been stored somewhere in the system), and increases the quality of the overall system configurations because it avoids unneccesary heterogeneity.
- However, not *all* parts of the rule can be easily checked automatically. The question whether, since the last graphics card purchase, a technology change took place, cannot at all be answered by a software system with low effort. Hence it makes sense to mark this rule as suggestive, but not mandatory. This is reflected by the fact that, in Figure 11 the suggested value "Matrox Mystique" can be accepted or dismissed, the final decision is up to the user.[16]

Consequences for the KnowMore machinery

So far, the example shows:

1. that knowledge objects can contribute to decision support in different ways: by offering supporting information, or by suggesting possible decisions – this must be indicated in the knowledge description and must be reflected by the way of presenting / processing the support information at the user interface
2. that different kinds of knowledge objects require (or, allow for) different kinds of processing within the knowledge brokering level, and at the application level (in the user interface) as well: while text or multimedia information might only be displayed in the user interface (if this is wished by the user), automatically processable knowledge

[16] By the way, it should be noted that the price "320" is also the price of the suggested product.

> items such as formal rule might be executed if the appropriate execution machinery is available, and then may lead to direct actions in some application system (like inserting a computed value in an interface mask).

Starting a new purchasing process

Now that we explained the major elements of the KnowMore variable editor, let's go on with a stepping through the application example. Assume the user "Andreas Abecker" has started a new purchasing process instance and has specified that he needs a new "Graphics Card". When filling the respective context variable, the system might be able to perform some easy Natural Language Processing (NLP) to identify that this natural language expression might be a linguistic representative of a corresponding entity "Graphics_Card" in a domain ontology that describes the classes of goods to be purchased at DFKI.[17] Some steps later in the workflow, a Senior Software Engineer in the IT infrastructure group of DFKI might be allocated the task to further specify the demand details. In order to do this job, he or she is provided with the variable editor for this specific task as shown in Figure 11. At the moment when the employee is confronted with the task – represented by this instance of the variable editor – the KnowMore system has already proactively evaluated the premodelled task-specific support demands, but does not yet present the results. Instead, it is stand-by with those results, for the case that the user wants them to see.

Information support is offered in the Information Browser

Now assume the user really wants some support information and presses the information button "I" in the variable editor. This results in opening the information browser shown in Figure 12. This browser offers the results of all queries to the organizational knowledge base as evaluated by the Information Retrieval agents of the KnowMore KBL, as specified at process modelling time and dynamically instantiated at runtime. This means concretely, e.g., that we could

[17] An easier solution for mapping natural language expressions to ontologically well-founded terms is to just offer a pull-down menu to the user for specifying his or her purchasing request where only ontology concepts are available in the menu. However, for a real-worl application, mapping NLP would be rquired – which is out of the scope of this thesis. Nevertheless, there have been promising experiments for mapping natural-language expressions to "ontologically-backed" expressions in several research areas [Budzik & Hammond, 1999; Budzik & Hammond, 2000] which also led to recent product developments like the OntoOffice tool of Ontoprise GmbH that – when working conventional Microsoft Office tools – links on-the-fly to semantically marked-up background information. There are also promising results in the area of Web Information Retrieval through combination of Natural Language Processing and ontological information [Guarino et al., 1999].

have had at process modelling time query templates such as, e.g.:

Example query templates which represent support requests

a) "search for yellow page entries about people knowledgeable in purchasing products of class X or one level more general in the product hierarchy"
b) "search for magazine articles or test reports for products of class X, including specific sub-classes or instances of this class X"
c) "search for evaluation reports comparing different products belonging to the product class X"
d) "search for recommendations regarding products of class X or the respective major product group which contains X."

Dynamic task-context allows template instantiation

These query templates could now at runtime be instantiated as follows:

a) "search for yellow page entries about people knowledgeable in purchasing a GRAPHICS_CARD or one level more general in the product hierarchy"
b) "search for magazine articles or test reports for GRAPHICS_CARDs, including specific sub-classes or instances of GRAPHICS_CARDs"
c) "search for evaluation reports comparing different GRAPHICS_CARDs"
d) "search for recommendations regarding GRAPHICS_CARDs or the respective major product group which contains GRAPHICS_CARDs."

Now, when evaluating these instantiated query templates, the retrieval agents in the Knowledge Brokering Layer may use ontologically represented background knowledge to further refine those queries, as follows:

Ontologies allow a refined query evaluation with background knowledge

a) "search for yellow page entries about people knowledgeable in purchasing a GRAPHICS_CARD, or PC_CARDS in general"
b) "search for magazine articles or test reports for GRAPHICS_CARDs, including S2, MATROX-MYSTIQUE, VOODOO-2, REVOLUTION-3D"
c) "search for evaluation reports comparing different GRAPHICS_CARDs"
d) "search for recommendations regarding GRAPHICS_CARDs or PC_CARDs in general."

Evaluating such queries yields the retrieval results shown in Figure 12, each results represented by a short textual characterization and then linked via a hyperlink to the respective original information source.

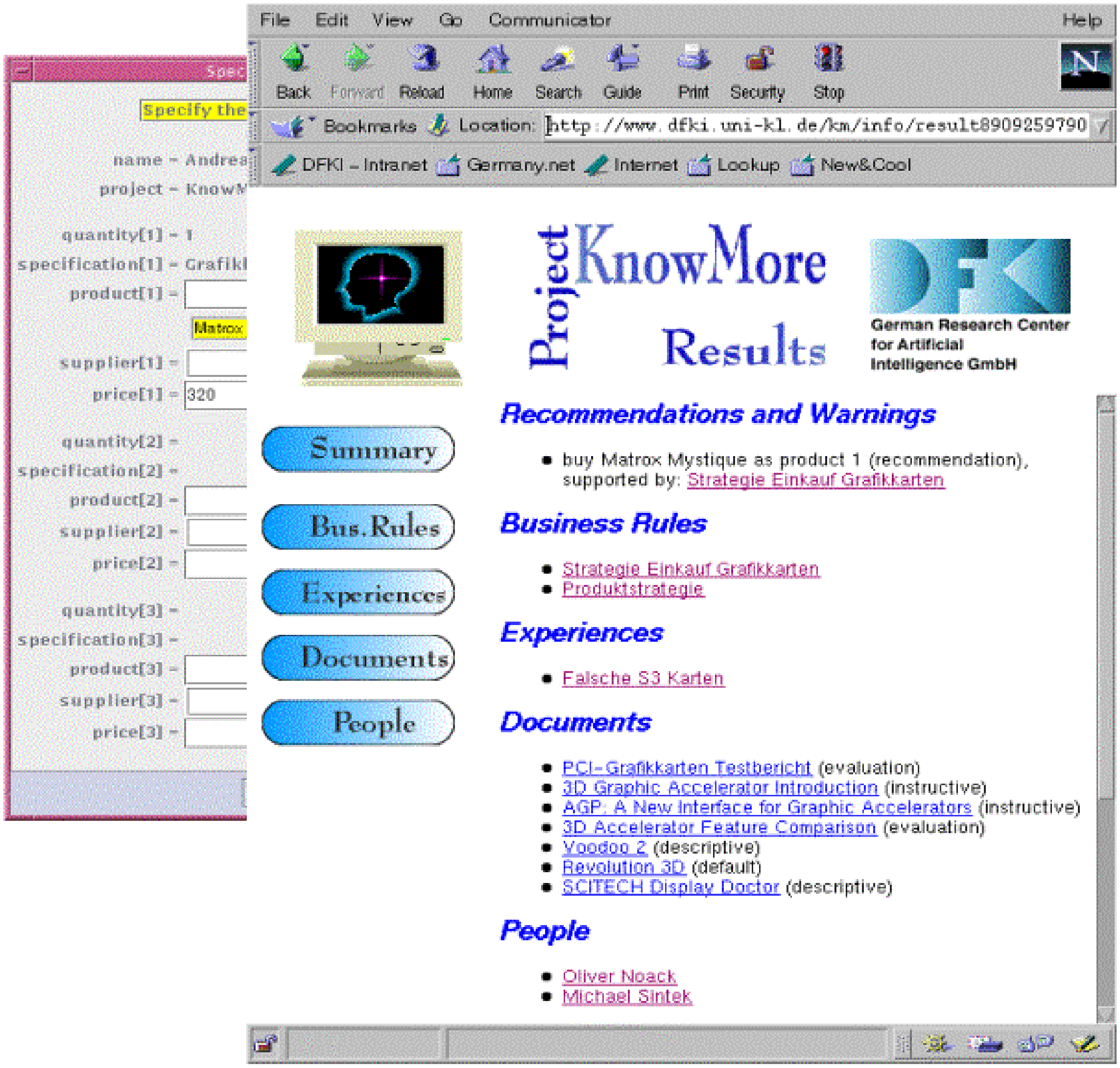


Figure 12: KnowMore offers Context-Sensitive Support

Intelligent ordering of results

In the information browser, the results are sorted and presented in a way which can be specified as part of the knowledge to be represented and executed at the Knowledge Broker Layer. In this example, we defined an order to be computed with the values of several metadata attributes represented in the Knowledge Descriptions (such as type of knowledge, reliability of knowledge and costs of knowledge usage) which were then aggregated in some notion of “importance” which corresponded to sort of “message type” as used for naming the respective sections of the information browser. These message types are shown in Table 11 below.

Kinds of information support and their ordering in the Information Browser

Message type	Explanation
Recommendations & Warnings	If there are entries in the knowledge base which allow to compute a suggested value or which are marked as message type «Warning». Will be presented with highest priority.
Business Rules	Entries in the knowledge base which are marked as «Business Rules» have by definition a high level of mandatoriness.
Experiences	Are, e.g., knowledge objects of type «Lesson Learned» or magazine reports with lesson learned content. Typically refer to a specific product. Are considered potentially more useful than general articles (next class below), because they describe concrete, justified experience, not only context-free knowledge.
Documents	All kinds of texts in the Internet or in magazines and journals. Relatively low level of importance (compared to classes above), because they represent relatively broad, not company- and application-specific knowledge.
People	Of course, personal experience and skills is in the normal case much more worthful than documented knowledge on paper and websites. However, in this application case we assume that the problem to be solved is first and foremost the job of the person to enact the given task at hand who should try to solve it alone as far as possible, and not to consume the time of other, expensive employees. So, access to people is ranked low in the list, in order to indicate that this knowledge source should be consulted for that job only as a last resort.

Table 11: Order of Message Types in the Purchasing Example

Orderings are declaratively expressed and computed on-the-fly

Of course, it is obvious that such ordering considerations are heavily disputable and clearly depend on the concrete application domain, the given information sources, maybe even on the experience and personal working style of the actual end user. Hence the presented example shall not be a "solution", it shall rather show that a full-fledged OMIS should provide (i) the means to express the respective matedata for formulating such heuristics as indicated in Table 11 and provide (ii) the mechainsms to process such ordering heuristics.

Accepting a suggested value

Let us assume the IT infrastructure engineer accepts the computed value "MATROX_MYSTIQUE" shown in the Figure 11 by clicking the "accept" button. This decision will then change the status of the respective context variable which in turn triggers an updating request for the information agents in the KnowMore KBL: since the results of information retrieval depends on the values of context variables, changes of context variables must cause a re-computation of retrieval results. The effect of such a context-sensitive re-computation as an answer to the acceptance of the suggested variable value "MATROX_MYSTIQUE" is shown in the transition from left to right (or, in the current orientation of the figure, from bottom to top) in Figure 13.

Context-dependent re-evaluation of retrieval results

This considerably narrows down the search in the OMIS. Now, no document occurs anymore which is about other specific products or has no direct relationship to this specific graphics card. What remains are:

i. compulsory purchasing business rules; and
ii. specific information about the "MATROX_MYSTIQUE" product.

If we would ask now for information support concerning potential *suppliers*, the system would yield only information about suppliers which are known to sell the "MATROX_MYSTIQUE" product, whereas in the previous process state (lower part of Figure 13), all suppliers would be described which sell graphics cards in general.

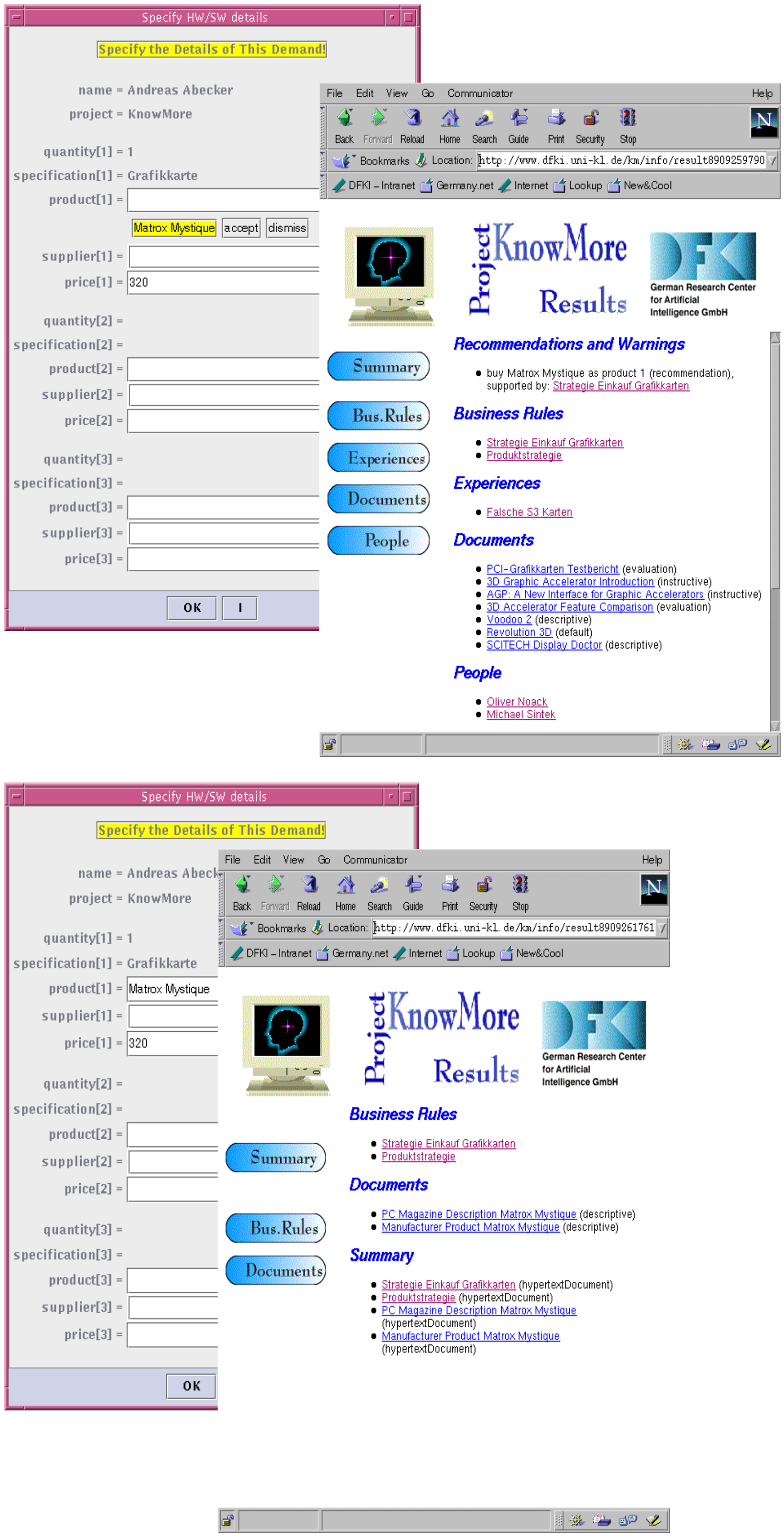


Figure 13: Context-Aware Information Supply

2.3 The KnowMore Contact Management Application

2.3.1 Process analysis

Sketch of contact management workflow

Since we discussed the purchasing example in much detail, we may present the next application example a bit more superficially.[18] Let us consider a very simple process: a first contact of a potential customer with a research institute as sketched (in a slightly simplified version) informally in Figure 14, together with potentially useful knowledge support offers.

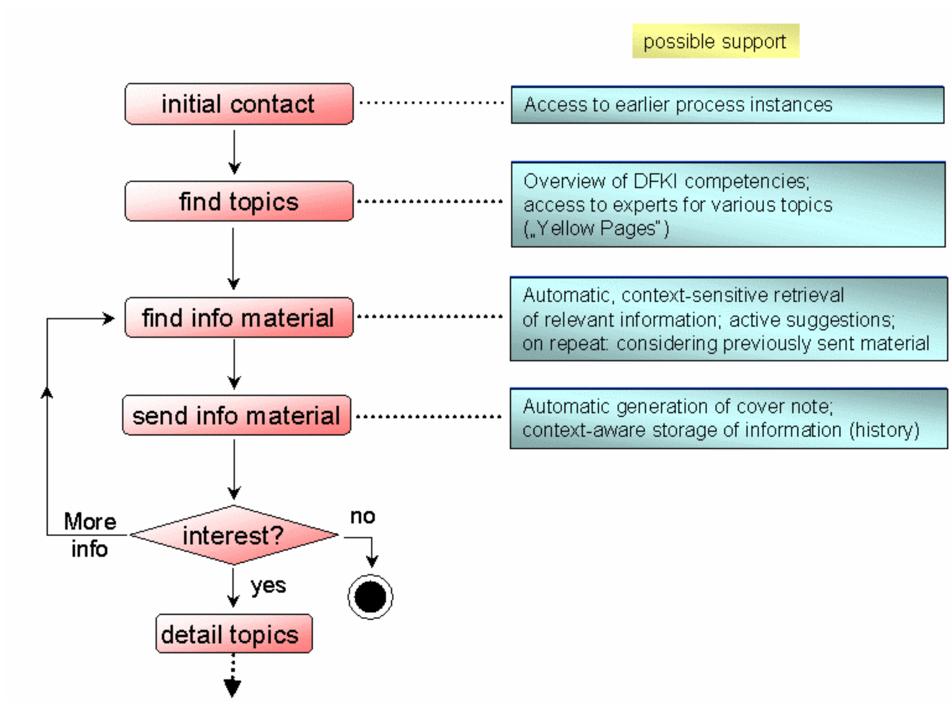


Figure 14: Contact Management Workflow

Initial contact: "rough demand specification"

The process starts with an initial contact which is typically a telephone call coming into the research institute, DFKI in the example. In this step, it is of utmost importance that the person receiving this call has immediate access to potentially existing information about earlier contacts to the same company (or even to the

[18] We described this application example the first time in [Abecker et al., 1999].

same person) worked on earlier by himself, by his department, or by some other department in the research institution.

Find topics: "detailed demand specification"

During this telephone call, or in a next step, as a wrap-up of the chat, the relevant topics of interest are identified wich might be important for the potential customer. For instance, in the telephone call it would be normal that the potential customer talks about his specific problem or area of interest, which is usually not directly related in a one-to-one way to specific offers of the research institute; but there are always vague collaboration possibilities which must be investigated a bit more. Normally, there may exist specific technologies or tools which might be useful for the customer, or there might be former projects dealing with similar issues as the customer's problems. The bigger the research institution is, the broader its range of offers for research cooperations, and the shorter the typical employee "life-cycle" time, the more difficult it can become to perform this task of finding the best match between existing tools, methods, technologies, etc. and the potential customer's concrete problem at hand; this may even be the case for a department manager who should be technically adept in at least one technology area.[19]

Find information material

Once it is clearly specified about which topics the potential customers should be informed, the next step is to compile an information package to send him – usually by mail, but maybe also electronically – which could give him some background information in order to assess the option of further following the contact, e.g., by a personal visit. Such an information package typically contains some fixed elements (like the institute's image brochure, or last year's annual report) plus specific topical information, for example, some technology whitepapers, a brochure about a specific tool, or project flyers. Having done this, a nice information package can be sent to the potential customer.

The process may go through several iterations

After some time there will be a reply: *No interest* stops this particular process, a request for *More Information* loops to the identification of relevant information material, while *interest* leads to further steps, e.g., the arrangement of a personal meeting or the definition of an appropriate offer to the customer.

---

[19] Even in the presence of today's sophisticated Customer Relationship Management (CRM) systems, this problem – which is fortified by increasing speed of technology development and business hypes, as well as by the arise of more and more virtual organizations and outsourced expertise providers – is not smaller than at the time of our first presentation of this example, but rather becoming bigger. Hence, we are still working on such topics [Hefke & Stojanovic, 2004].

Proactive support by the KnowMore system

Most of the activities in this process can be considered knowledge-intensive, and their support by pro-active information delivery from an OMIS can profit much from a meaningful use of context variables: When selecting the information material to be sent, active suggestions from the system would be helpful, supposed that the system takes into account the information from the activities done so far, e.g., the selected topics.

What we will see later: In this process, an automatic, **context-aware archiving** of results is useful when a similar process is started at a different time and / or location: The process step "initial_contact" will then profit from information about earlier contacts to the same company, or about similar cases.

We summarize relevant sources of information in OMIS as listed in Table 12. This table indicates that in this process – where information processing is the centre of all activities – it might occur that it becomes difficult to distinguish between operational activity to be done at the "business level" and support activity at the "knowledge meta level". This is essentially because the "business" is a "knowledge business" (cf. remarks, marked with an (*)). This is not untypical when dealing with knowledge-intensive activities and should thus not lead to confusion.

Type of knowledge source	**Content and usage characteristics**
Earlier process instances	- Represented in the OMIS as indexed summaries of process instances, with links to created documents from those process instances - Especially high importance if there was a contact to the same company, in particular in the first process step - There may be confidentiality issues in a multi-department organization
Organization competence map	- An explicit knowledge map visualizing the competence ontology of the organization in order to support navigation between topics in the step "find_topics"
Image brochures	- Brochures for the organization or for specific departments belong to the standard material to be considered because it is sent to potential customers by default - Hence these are not so much a source of *background*

Type of knowledge source	**Content and usage characteristics**
	knowledge for the employee, but rather *operational* objects to be dealt with when enacting the process (*)
Flyers and leaflets	- Typical public relations material for projects, solutions and competence areas - See remark above: (*)
Technical documentation & scientific papers	- The more specific a concrete customer request is, the more relevant might be technical information sources describing solutions and results achieved earlier - This kind of information might play a twofold role: (i) for internal use in order to clarify what offer could be made to a potential customer; (ii) as reference material for a potential customer, in the sense of (*) above
People	- Homepage / yellow page of people who are contact persons for specific projects, topics, or departments

Table 12: Relevant Information Sources in the Contact Management Application

2.3.2 Runtime Support by the KnowMore System

Using the prototype system

Figure 15 shows a screenshot of our system prototype. On the left, in the background, we see a KnowMore variable-editor window used to represent the process step “find_info_material”. After initial specification of the topics of interest “KNOWLEDGE_MANAGEMENT”, “KNOWMORE” (a specific project), and “ESB” (another project), we observe the system answer in the variable editor and the respective information browser shown in (Figure 15) and described below.

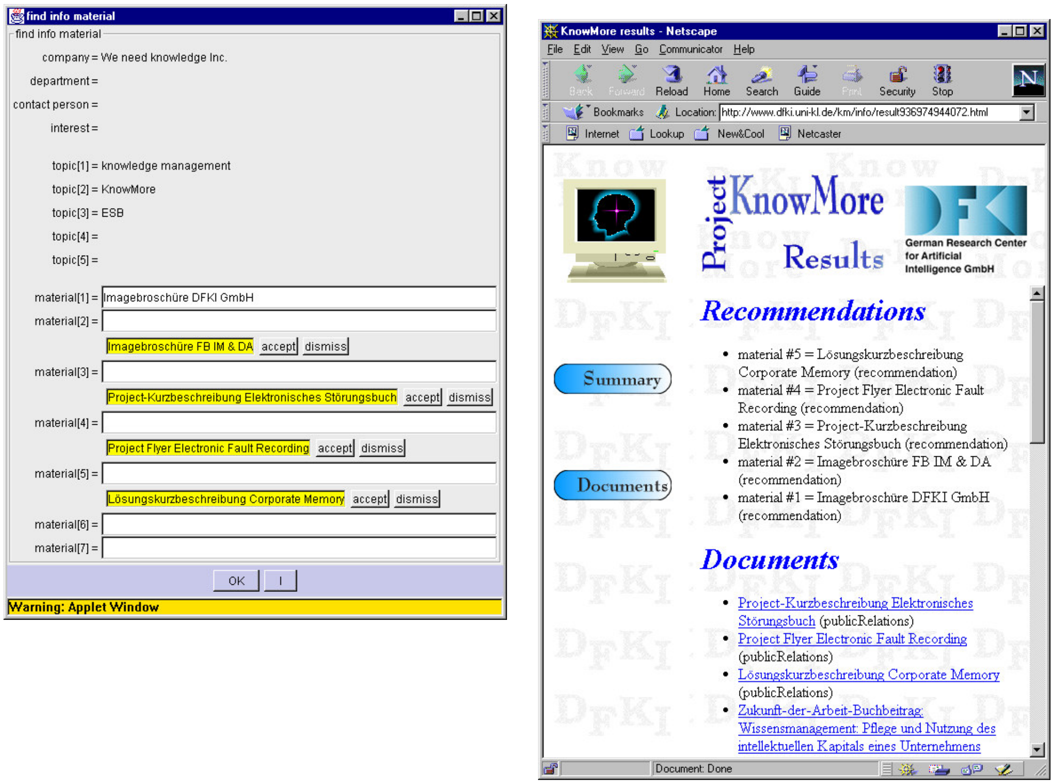


Figure 15: KnowMore Knowledge Support in the First "Round"

Suggested values in the variable editor

1. In the variable editor (left hand side of the figure):
    1. As a first information material to be sent to the potential customer, the "DFKI Imagebroschüre" has been specified automatically. This computed value is a mandatory part of all information packages to be sent, and has thus directly been inserted as the value of the variable MATERIAL[1].
    2. As a further suggested value, the image brochure of the most relevant department has been proposed for MATERIAL[2]. This has also been specified as a default material, but with a lower degree of bindingness, because there might be cases where it is more appropriate to present the organization as a whole to a potential customer.
    3. Then, three project flyers and leaflets have been found and are suggested for inclusion in the information package. They have been found because Conceptual Information Retrieval computed their

relevance with respect to the topics of interest specified in variables TOPIC[1] ... TOPIC[3], and they are considered highly relevant, because the retrieval agents in the given application can exploit the metadata and "know" that this type of document is particularly suited for customer contacts.

Further information in the Information Browser

2. In the information browser (right hand side of the figure):
    4. Besides the suggested values for material to be sent to the potential customer, additional scientific papers and technical documents are listed which – because of their conceptual index somehow related to the specified topics of interest – might be interesting for the actual employee, either to further work out a potential technical offer, or to be sent also in an information package.

The user has always the control

As it was already mentioned in the example before, all suggested values are subject to acceptance or rejection by the end user. Whatever decision he or she may reach, in the next step ("send_info_material") a cover letter to the potential customer can be prepared (partially) automatically from the data already known.

Document storage with process-specific metadata

Further, this letter can be stored in the OMIS, together with a conceptual index stored in the KDL which describes the situation where this letter has been prepared in (a contact with a representative of company X, interested in topics Y, Z, U).

If in a later contact to the same company, or in a next iteration of the same process instance, there would arise the necessity to send another information package to the same person, it could be taken into account that there has already been an earlier information package with a certain content. Those considerations are reflected in Figure 16:

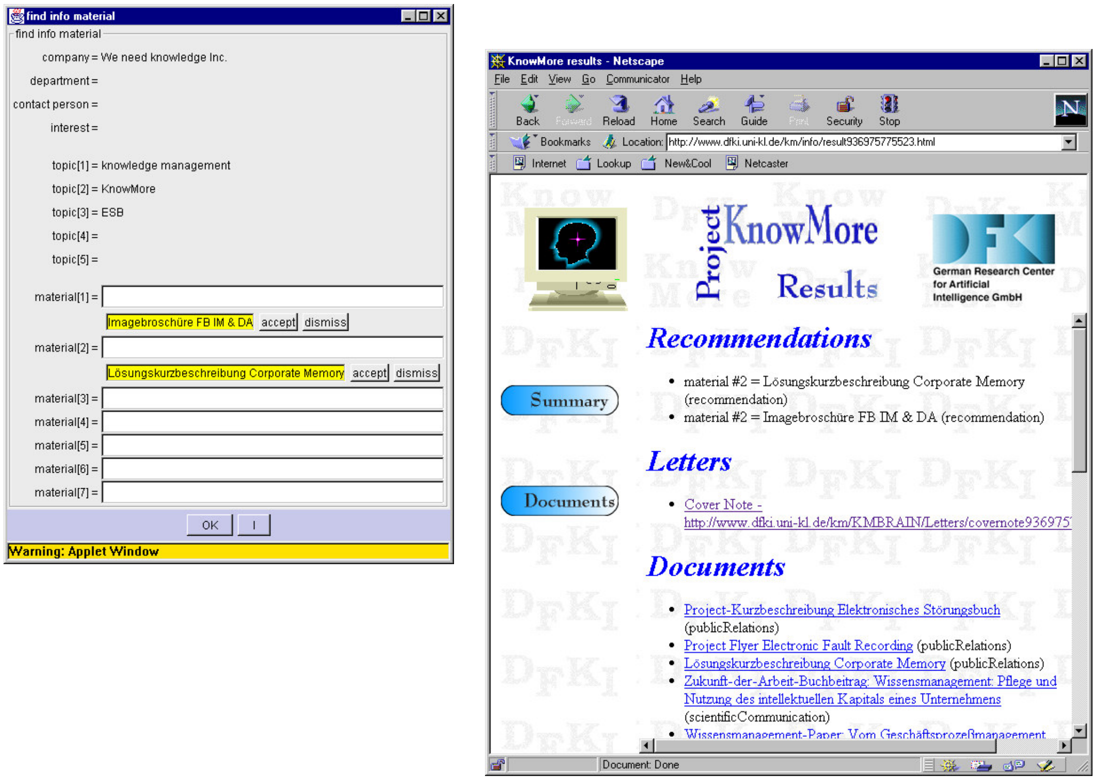


Figure 16: KnowMore Support in Second "Round"

Dynamic task-context in iterative processes

If we enter the activity "find_info_material" for a second time within the same process and after a relatively short amount of time elapsed, the information agent evaluating the associated information requests, can:

- find the already sent letter in the archive: this letter is now shown as an additional information offer in the information browser;
- anaylse the metadata index in the KDL which describes in detail the situative context in which the last letter and information package has been sent;
- upon this analysis, leave out the suggested material already proposed and sent in the earlier mailing (in the example, this concerns, e.g., the DFKI Image Brochure which was mandatory in the first mailing to the potential customer);
- and consequently add new information material which may extend or be more specific than the material already sent.

The role of the two introductory KnowMore examples

As a side remark it should be noted that we do not consider such services themselves as completely earthshattering and only achievable with our and only our solution. However, we want to show the possible benefits of information systems which maintain a dynamic task context to be supportive of information supply services, and to show the potential of explicitly storing and manipulating representations of such context descriptions. We have to use, of course, an example which is *as simple as possible* not to draw off the attention and cognitive load from the essential ideas, but *complicated enough* to demonstrate the ideas in a more natural way than a “toy example”. The major contribution of this thesis is not so much to show that such services are realizable *and how*, but rather to point out the principles, ideas, concepts, and fundamental approaches in order to easily build systems with services as the ones demonstrated. This is also the purpose of the following short section about implementation.

2.4 Implementation of the KnowMore Prototype

This section describes the implementation of the KnowMore prototype as demonstrated with its functionality in the preceding Sections 2.2 and 2.3. The purpose of this section is not so much to define an authoritative way of implementing such a system, it shall rather give an idea that functionalities as the ones described can be implemented in a scalable manner, on the basis of software architectures fully compliant with existing standard approaches.

First of all, we can distinguish between tool support for process definition time (modelling time) and for process execution (enactment).

Server tools for system set-up and maintenance

2.4.1 Tools for Process Modelling Time

For process definition time, the following elements are required:

- A **business process / workflow modeling tool** which is able to define process models which are later on executed by the KnowMore workflow engine. This modeling software must also allow to define extended process models which contain demand specifications. For our experiments we used the ADONIS commercial Business Process Management tool. The KnowMore-specific extensions were modeled as comments in activity descriptions, and then appropriate parsed and executed by our Workflow Engine.
- An **ontology modeling tool** in order to formalize conceptual structures underlying the ontology-based information modelling in the KDL. To this end, we just used a text editor for editing in ASCII code the KnowMore knowledge models in the KnowMore knowledge representation language OCRA used at the begin of our experiments. Of course, arbitrary ontology modeling tools such as Protégé[20] or KAON[21] could be used to this end.
- An **annotation tool** for describing OMIS knowledge sources with ontology-based metadata as required fot the KDL.

[20] http://protege.stanford.edu/

[21] http://kaon.semanticweb.org

Figure 17 illustrates the cooperation of these tools at process definition time as it was implemented prototypically in the KnowMore project. Two other elements which are not in the focus of this thesis, but gave rise to other interesting research and could be important for KnowMore-like approaches in a broader practice, are also showed in the overall picture:

Semi-automatic ontology construction

- Building domain ontologies in complex and maybe frequently changing application domains from scratch and by hand, can become cumbersome and error-prone. Hence our work on ontology-based Information Retrieval initiated some early work on text-based, semi-automatic acquisition of ontologies by statistical text analysis methods (in Figure 17 indicated by the Trex Similarity Thesaurus Generator tool). In the meanwhile, the topic has evolved into a mature research area which produced already impressing results (see, for instance, [Maedche, 2002]).

Modelling facilitator tools at process definition time

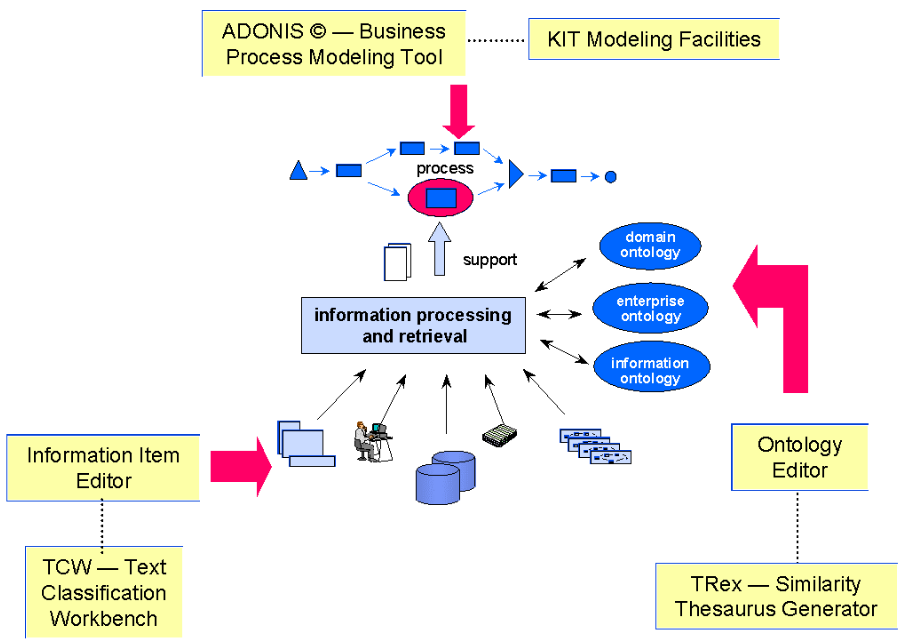


Figure 17: KnowMore Tools for the Process Definition Time

Semi-automatic metadata creation

- Another topic which can represent a significant hurdle for introducing KnowMore-like systems in practice, is the effort for annotating OMIS content with ontology-based metadata (in the meanwhile, this topic has become prominent as the "Annotation Bottleneck" in the Semantic Web area). Consequently, investigation of automatic, learning text categorization tools was a natural part of our research. While our first experiments go back for a long time (see, e.g., [Tschaitschian et al., 1997; Abecker et al., 1998e; Junker & Hoch, 1998]) there have also been developed mature results in the recent past [Junker, 2001; Handschuh et al., 2003].

2.4.2 Tools for Process Enactment: KnowMore Server

Now we come to the software components required for running the system at process execution time: We discuss the different parts of our implementation going through the several architecture elements depicted in (the slightly simplified architecture shown in) Figure 18.

First, we can mention that KnowMore has been implemented as a JAVA-based system that realizes a classical Client-Server (C/S) architecture.[22]

The KnowMore server

The KnowMore server hosts the following elements of the system:

(A) All relevant data and knowledge bases;

(B) The KnowMore workflow engine;

(C) The KnowMore information agents, in the figure comprised under "Knowledge supplier" which interacts with the "Inference Engine";

(D) All software for process definition time and system maintenance.

(D) has already been described above. Let us discuss the remaining components with some more detail:

Data and knowledge bases kept on the KnowMore server

(A) Static data and knowledge bases: the KnowMore server holds the following data, knowledge and information sources:

- The business process models (to be enacted by the KnowMore workflow engine, including information about organizational structures, roles of employees, flow of tasks), extended by support specifications (generic OMIS queries) plus KIT variables, i.e. dynamic context variables for Knowledge-Intensive Tasks.
- The OMIS content in the narrower sense, i.e., personnel yellow pages, technical documentation, product data sheets, corporate regulations, etc.

[22] In order to show how the KnowMore approach fits into standard software environments, we mark with blue and green colour in the picture all elements which occur already in the Workflow Management Coalition's (WfMC) reference architecture and which are implemented unchanged or could be implemented pretty similarly following the WfMC's recommendations. In contrast, yellow colour marks all elements which have been added to realize the specific KnowMore functionalities.

- The Knowledge Item Descriptions, i.e., the content of the KDL: formalized descriptions of content and context of concrete information sources and information items.
- The ontologies, i.e., Information Ontology, Enterprise Ontology, and Domain Ontology(ies) which provide the vocabulary for knowledge item descriptions and the background knowledge for intelligent retrieval services.

The KnowMore workflow engine (WFE)

(B) The KnowMore workflow engine (KnowMore WFE) interprets business process models. To this end, workflow-specific databases are created and maintained for workflow-control data and workflow-relevant data (here, we follow the approach and terminology of the Workflow Management Coalition [WfMC, 1995]). The KnowMore WFE has been implemented as a simple JAVA program that interprets process models, role and enterprise models, and dispatches tasks to the appropriate person(s) in the organization.

Knowledge supplier and inference engine

(C) Also on the server, is the complete machinery required for task-specific, context-dependent information delivery. This is realized by a software module called “Knowledge Supplier” in Figure 18 which is another JAVA component that instantiates generic OMIS queries (as given in the support specifications of knowledge-intensive workflow activities) with concrete, actual values of context variables (held in the database for workflow-control data mentioned above), and then evaluates these instantiated queries by inspecting the knowledge item descriptions explained above and retrieving potentially relevant descriptions. Since all knowledge item descriptions are formulated in OCRA, the KnowMore specific object-oriented knowledge representation language (see Appendix), the retrieval process employs the OCRA inference engine.[23]

---

[23] The OCRA formalism (Object-Centered Relational Algebra) has been designed – based upon our requirements analysis for the KnowMore KDL – and implemented by Michael Sintek and was used throughout the KnowMore project for representing and processing knowledge. OCRA is an object-centered representation formalism with a query evaluation module implemented in JAVA that translates OCRA queries into relational database queries (the RDBMS for data storage is coupled with JAVA via the JDBC standard interface [JDBC, 1998]. The system shall combine expressive power and effective inferences exploiting class hierarchies, subsumption, and transitivity of inheritance, with mass data storage, transaction principles and industrial-strength implementations of relational databases. It was also used in a commercial knowledge management project for the conservation and analysis of maintenance experiences with complex machines in black-coal mining [Bernardi, 1997; Bernardi et al., 1998]. Technically, OCRA classes are mapped to relations, and objects to tuples, while embedded objects are represented by their object identifiers. Of course, similar

2.4.3 Tools for Process Enactment: Client Side

The KnowMore clients

On the client side, i.e., on each logged in user's individual PC, we have personal worklist handlers, implemented as JAVA applets which connect to the server via standard TCP/IP sockets. The communication protocols are designed mostly according to the WfMC standards.[24]

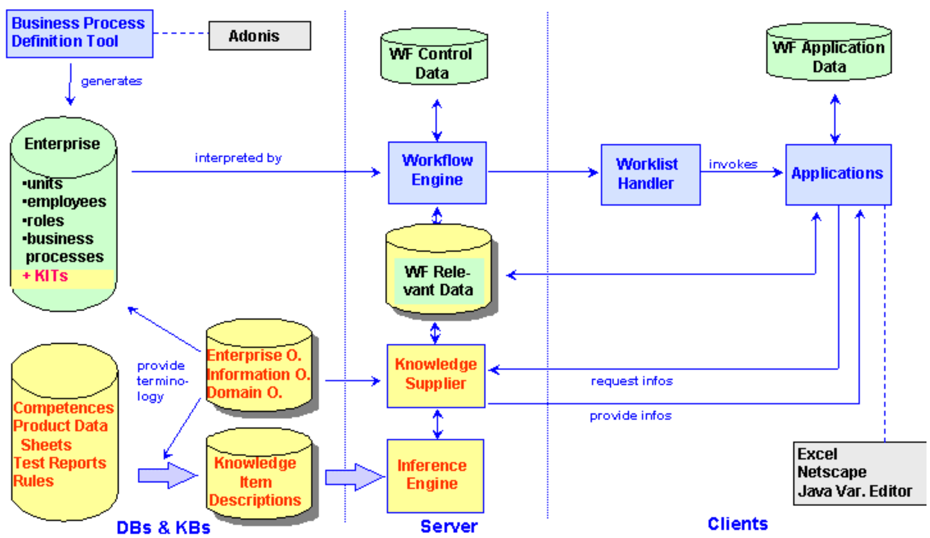


Figure 18: The KnowMore System has a Web-Enabled C/S- Architecture

Worklist handlers offer open tasks in a role-dependent manner

A worklist handler offers to the user a list of tasks actually assigned to him / her for processing. The task assignment is done according to the organizational role of the user and the respective information about role assignment and organizational model in the process model. Of course, the same task might be assigned to several people in the organisation if they enact the same role (for instance, a department might have two secretaries without a clear task separation, or a research

functionalities could have been realized by using Object-Oriented Database Systems (OODBMS) or Object-Relational Databases. Today, storage capacities and processing power of some ontology management systems (like KAON [Maedche et al., 2003]) could also be sufficient for non-trivial KnowMore-like applications.

[24] Though this was not the major goal of this project, we nevertheless showed that in principle, the workflow-specific parts of the KnowMore architecture could also be realized by commercial tools. We investigated this question a bit deeper in [Abecker et al., 2000c].

departments might have 10 researchers with approximatley the same qualification and knowledge areas). In this case, it is up to the users who will take-up the open task first. When one user starts to work on a given task, this task is, of course, deleted from the other employees' task lists.

When starting to work on a task ...

In the moment when a user opens a task from his or her worklist handler, two things happen:

... the appropriate application software is started, ...

1. According to the task specification in the process model, the appropriate tools for performing this task are started. Commercial workflow software which acts as a powerful middleware for coordinating user activities, information and document flow and other software tools, has interfaces to a whole bunch of application software which is then started automatically with the appropriate data and documents. For our simple demonstration scenario, it was sufficient to simulate those application software tools by our KnowMore variable editor already introduced in the previous sections.

... and the respective Information Agents are activated

2. In parallel to the application software, the appropriate Information Agents are started that evaluate the modelled support specifications associated with the given task. In Figure 18, this is indicated by the "Knowledge Supplier" module hosted at the KnowMore server which comprises the application code of the several Information Agents realizing specific query evaluations. These Information Agents continuously communicate with the application software (practically speaking, this is currently only the KnowMore variable editor which makes this task technically much easier) in order to monitor user behaviour and get notified of changes in context variables which might have an influence on the given retrieval tasks. Further, they also update continuously the information offers to be shown in the Information Browser or to be implanted in the user's application software as suggested values.

Information Agents monitor user bevahiour to maintain an up-to-date context model...

... and to update context-specific information offers

2.4.4 Processing Information Needs

Now, let us have a look into the internal processing of information needs within such information agents which implements the "intelligence" of the KnowMore system and provides the most innovative software functionalities.

Figure 19 shows how several kinds of represented knowledge interact in order to fulfill an information need when an information agent answers a query. Figure 20 instantiates this scenario by a concrete example.

The chosen presentation particularly separates out the several processing steps and sources of background knowledge that are typically mixed together and intertwined in human information gathering or specialized performance support systems. The aim of this component-based view is to suggest the way towards a generic, widely usable, modular software architecture on the basis of integrated processing of formal and retrieving of informal knowledge.

Information needs are satisfied by integrating inferences over formal and retrieval of informal knowledge

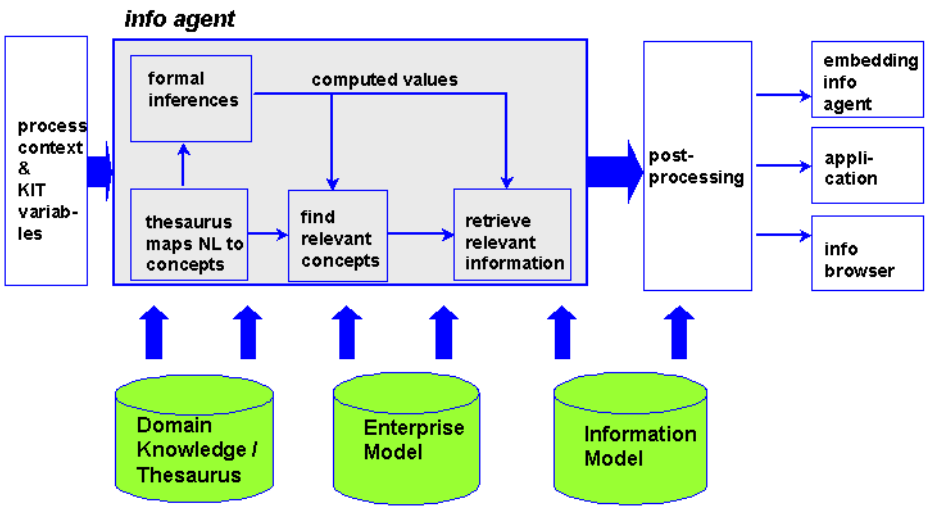


Figure 19: Processing Information Needs

Basically, an Information Agent can become active in a certain situation if the modelled preconditions are fulfilled. To check these preconditions and answer the associated query it has to have access to:

Input processed for checking preconditions

- the especially created KIT variables, or dynamic context-variables (Cp. Table 7: *Which kind of product shall be purchased? What concrete product? At which price? From which supplier?*), and
- to the global and local process context *(Overall business task, activity performers and their respective roles, time conditions, etc.).*

Simple preconditions, ...

In the KIT example in Table 10, we used only simple conditions, e.g., whether there already exists a value for a given KIT variable or whether such a value is in a certain class of the domain ontology, or not.

... more complex preconditions, ...

A more complicated precondition would be, e.g., to detect an important purchase depending on the sum of the expected prices, or on the type of goods to be bought. One could also imagine preconditions which are evaluated using local and global process parameters (e.g., whether a given purchase is considered to be important, can also depend on who initiated the purchase).

... most complicated preconditions

As another example take the delivery of pointers to knowledgeable colleagues by querying a skill database / yellow-page system. Since these colleagues may spend their time for helping the actual user, such an information service might only be appropriate if the actual user is unexperienced. Finding out whether this is the case or not could be done by another information agent which, e.g., seeks for other similar purchases performed by the same employee, or looks up when this employee started working in the company.

If preconditions are fulfilled ...

Now, we go further into the details of the information agent's core functionality. To this end, we have a look at Figure 20 which shows an instantiated version of the processing schema presented in Figure 19 and how it would be handled by the appropriate information agent. Essentially, we can identify three main processing steps:

... three core retrieval activities can be started

1. Map application situation onto retrieval concepts.
2. Perform knowledge-based query expansion.
3. Retrieve information from various sources.

We discuss these steps in more detail:

(1) Map application situation onto retrieval concepts.

Since our system directly takes its query input from the application program (e.g., from the product specification editor), it cannot be guaranteed that the user / employee filling out a demand form exactly uses the ontology concepts which organize the knowledge archive. Thus, we have a thesaurus system linked together with the domain ontology which ensures that other synonymous or similar terms possibly used in the application can be mapped to the appropriate query concepts.

Currently, the thesaurus information mainly deals with:

Linguistic ambiguities to be addressed at the user interface

- multilingual use (German vs. English),
- different writing (data base vs. database), and
- different naming conventions (terminological logic vs. description logic vs. KL-ONE-like system).

Lexical ontology layers as a solution approach

In principle, a full Natural-Language Processing machinery could be employed for this purpose. As already mentioned earlier, combined linguistic / thesaurus and ontology methods have been successfully tested in a number of Information Retrieval applications (e.g., [Guarino et al., 1999]). Today, advanced ontology management systems like KAO N provide a *lexical layer* to represent the relevant knowledge for dealing with such phenomena (cp. [Maedche, 2002; KAON, 2004]).

Alternative: browsing and navigation in ontologies

Of course, such problems could simply be resolved offering to the user a selector box which displays the available ontology concepts, instead of free text fields. However, integrating thesaurus-like structures – which maintain sets of evidences for each ontology concept – provides interesting perspectives for further developments:

- it is easier to use than navigating in complex ontological structures;
- it is also possible without any cooperation between application program and assistant system, because the information assistant could analyse the documents created by the application, or even watch the keyboard actions waiting for triggers which activate an information need;
- the approach works also in non-interactive scenarios, for instance, if customer error reports coming in per e-mail shall automatically be assigned to certain problem classes with their respective answer documents;
- in such a scenario, a browsing-like approach would be inappropriate for another reason, too: Customers and diagnosis experts often think in different conceptual structures and terms, such that also presenting an ontology-browser would not necessarily be useful.

To sum up, we see that there is both a significant application potential and some promising realization methods. Of course, in this area, more application-oriented research is required.

(2) Perform knowledge-based query expansion.

While the above first step is concerned with a potential *terminology* mismatch between application or user language and query vocabulary, the second step deals with matching query concepts with index concepts used in the repository.

Ontologies may help to overcome the core problem of Information Retrieval

Here, the core problem of information retrieval occurs: information needs are often only vaguely specified, without clear knowledge about what knowledge sources will really be useful; document indexing is uncertain as well, because documents are often „more or less“ relevant for specific topics in a given situation; moreover, it will often be the case that no document in the archive exactly matches the actual information need; in such a case a human information searcher would try to slightly *reformulate* the queries in order to find *some* answers to the „second best question“ instead of *no* answer to the best one.

Ontologies provide background knowledge for information search

Enriching, substituting or reformulating the query concepts is done in the second step. We assume that general, as well as task and domain specific **search heuristics** are needed which exploit the structures specified in the underlying ontologies. Nowadays it is commonly accepted that subconcept-superconcept relations of index concepts described in domain ontologies should be utilized to support precise-content retrieval in Digital Libraries [Welty, 1996] and OM systems [O'Leary, 1998], or for the Internet [McGuiness1998; Stojanovic, 2003a].

However, beyond this very general statement, most approaches use only very simple search heuristics (like, *„If there is no document about* `x`*, then search for a document about* `superconcept(x).`*“*), or rely on manual browsing through the ontology.

There is a need for sophisticated search heuristics

Though such general search heuristics may be valuable, we see a clear need for more powerful heuristics expressions to be evaluated at runtime, e.g., taking into account actual situation parameters:

Examples for different required usage of ontological background knowledge

- For instance, if you are searching for *business rules* concerning the purchase of a graphics card, all business rules about purchase of any superconcept (hardware, any good) are also applicable, but it makes no sense to look for a business rule about purchasing a `Matrox Mystique`.
- On the other hand, if you are looking for a *competent colleague*, anyone who bought any graphics card recently (a `Matrox Mystique` as well as a

`Matrox Milennium`) will have some basic experiences about graphics cards and purchases in general.

- However, if the performer of the actual workflow activity is a hardware specialist himself, it probably makes no sense to point him to another employee known to be competent in hardware questions, except the expertise of this colleague is more specific and better suited for the actual case.

These examples show that depending on the kind of investigated knowledge source or depending on query context parameters, pretty different interpretations of the same ontological structures might be appropriate.

Another sample application domain

Things get still more interesting when switching from our purchasing example to some other applications. Consider, e.g., the machine model of a complex technical facility as the domain of discourse used for indexing machine diagnosis experiences.[25]

Search heuristics might not only move up and down in the concept taxonomies, but also sidewards.

Here, when searching for observations concerning a certain machine part, it is often a good idea to take into account observations associated with another part of the same machine module, since there are mechanical and functional influences. From the query point of view, this means to search not only for the given concept, but also for other subconcepts of its superconcept in the `part-of` model of the machine (i.e., for sort of *siblings* in the machine model).

The analogy in our purchase domain would be to search not only for technical documentation about `graphics cards`, but also for material about `network cards`, which is nonsense in the general case, but could make sense in some situations, e.g., when searching for knowledgeable colleagues.

Coming back to the mechanical engineering case: Another example are electrical or hydraulic connections represented in additional models of the machine which are useful for query expansion in some cases (depending on what kind of machine failure is examined), but not in others. Here we would need some rule language to formulate search heuristics which allows to specification application conditions.

---

[25] See [Bernardi et al., 1998; Bernardi et al., 1998a; Bernardi et al., 1998b] for a fielded application of this idea, or also [Dengel et al., 2002] for a sketch of some ideas.

These examples show that simple generic search heuristics (i.e., the same strategy is used in all situations) are not sufficient for complex scenarios. In [Liao et al., 1999] we discuss how one could formulate search heuristics over domain ontologies in a way as it was implemented in a prototypical personal competence search tool for a research group.[26]

The instantiated example shows how several kinds of background knowledge interact.

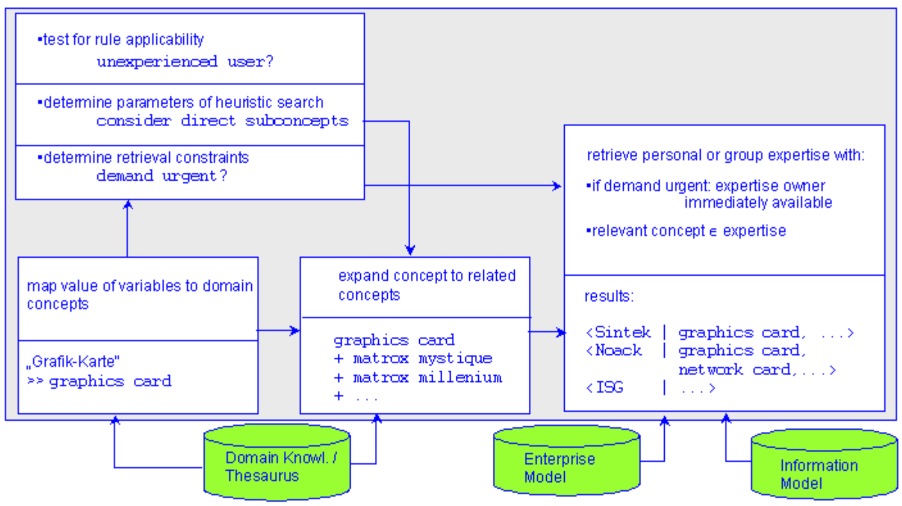


Figure 20: Instantiated Retrieval Example

(3) Retrieve information from various sources

The last step concerns retrieval in the narrower sense. At that moment, query concepts and query constraints (i.e. restrictions formulated over metadata like answer time, access costs, or information reliability) are put into a selection statement for the object-centered relational algebra OCRA. Our retrieval machinery basically realizes some deductive database functionality.

This retrieval functionality then delivers knowledge descriptions of possibly rele-

[26] We will come back to this topic later when discussing in detail the different layers of the KnowMore architecure.

vant sources. The knowledge descriptions specify how to access the content of these sources. In our demonstration prototype we just assume that all sources have a URL which can be linked into the information assistant's result HTML page.

Inferences for preparing retrieval

Again, we have an interaction of retrieval and formal inference, since values for ***query constraints*** can be formally derived, or delivered by embedded information agents.

For instance, if we have an urgent demand (this can be determined with the help of the global business process parameters) it makes no sense to list pointers to colleagues not immediately available. Whether some colleague can immediately be called, can in turn be determined (at least partially) by checking the databases for holidays and the business trips.

Inferences for post-processing results

During ***postprocessing***, for sorting out some, or at least for ordering the pointers to colleagues, the enterprise ontology can be taken into account. For example, it might be wished that people working in the same *project* are preferred over people only in the same *department* or at the same *site* of the company. It might also be preferable not to present people which are above the actual user in the organizational chart (because asking them costs more money than finding out the information by himself).

2.5 Summary

In this introductory chapter on the KnowMore approach, we gave a provisional, somehow informal (which means, not technically thorough), yet relatively detailed overview of the framework, two running examples, and the implementation of the KnowMore system.

Presentation approach of this chapter

The overwhelming amount of details of the examples and the implementation, which was nevertheless not a formal, complete and consistent, technical description might look somehow confusing for the reader. However, I think the major practical potential and scientific innovation of the KnowMore approach comes mainly from the integration of manifold bits and pieces from different ideas in AI and IT into one coherent architecture, and from the coordinated, powerful, and purposeful interaction of those bits and pieces. This can only be explained going into some level of concrete and exemplary detail that – on the other hand – demands at least a partial explanation of the overall approach and relationships and theoretical basics, in order to understand its functioning, and its innovative parts, as well.

Content of next Chapter

After this example-driven, illustrative part, we can go on in the next Chapter with a more scientifically sound and rigid presentation style, going step by step through all layers of the KnowMore architecture, discussing the major design rationale for each layer, its goals and functionalities in a generalized manner, and showing different possible interpretations and implementations of these layers. So, we come to an abstraction of KnowMore in the kind of a reference architecture which allows manifold different instantiations where the so-far presented KnowMore system is one of.

Before we do this generalization step, let us briefly summarize what we saw already in this overview Chapter.

KnowMore characteristics

- The major functionality of KnowMore is to provide **pro-active, task-specific information support** on the basis of a **dynamic context model.**
- In contrast to many other, Expert System oriented approaches which support only one task type with deep, heavy-weight techniques,

KnowMore rather aims at **supporting arbitrary (knowledge-intensive) business processes** in an organization.

- The price for this level of generality is that of a potentially **lower degree of system "intelligence"** (autonomous problem-solving functionalities in the system's services). This is **compensated by** a much **broader applicability**, a high level of **flexibility**, and a very **easy integration with existing applications and ways of working.**
- In general, the basic change (or, hopefully, advance) from traditional Expert Systems towards KnowMore-like functionalities is that of going **from an Automation System to an Intelligent Assistant System** (which provides its own, challenging and promising, research questions with regard to interface issues, optimal man-machine interaction, etc.).
- The basic technical hook to allow for those features (integration with existing environments, pro-activity, dynamic task context) is the integration with and extension of Business-Process Management approaches for modelling and analysis, and of as – technically – Workflow Management for enactment. This is the reason that I consider this approach essentially a support concept for **Business-Process Oriented Knowledge Management.**

For the methodological structuring and the technical realization of our approach, we propose a four-layer architecture:

1. The Knowledge Object Layer (KOL) represents the whole breadth of information in the Organizational Memory. Seen from the end user point of view and seen from the economic perspective of the end user organization, the major design decision is that we care about most **heterogeneous data, information, and knowledge in manifold, maybe semi-formal and informal representations and media.** This allows optimal exploitation of existing knowledge and information sources in the organization and keeps formalization costs small.
2. The heterogeneity and in-formality of the KOL necessarily causes the existence of a declarative, knowledge rich, meta-data oriented Knowledge Description Layer (KDL) which provides a **homogeneous** (technically and

conceptually) **access to heterogeneous sources,** which provides meta information about sources, establishes links, interconnections, and relationships, etc. We propose to realize this layer using formal ontologies.

3. The rich information provided at the KDL can be exploited in the Knowledge Brokering Level by **intelligent information retrieval and processing** methods. We shortly mentioned the idea of retrieval heuristics which will be discussed a bit more in the following Chapter, and we showed the complex possible inference patterns which can be realized through the collaboration of different Information Agents coordinated via preconditions and post-processing rules.

4. The Application Layer provides the interface to the end user which **integrates non-intrusive, yet pro-active knowledge services into the workflow-oriented, daily work environment.** Further it realizes a dynamic, local (to a task) and global (for the business process instance) context management for all system services.

KnowMore implementation

To some extent, we illustrated how such an architecture was implemented in the KnowMore system by discussing:

- The <u>KnowMore server</u> holding all relevant data and knowledge bases, as well as metadata, background knowledge, the workflow engine and the Information Agent software.
- The <u>KnowMore clients</u> on which the task-specific application software, the Information Browser and the dynamic communication mechanisms between Information Browser and server-hosted Information Agents run.
- The <u>support machinery for process analysis and definition time</u> which is useful (and, maybe, necessary) in order to run a KnowMore-like scenario in practice, i.e., process modelling tools, ontology engineering tools, support for text-based ontology engineering, and (semi-)automatic text classification for metadata creation.

In general, the chosen application architecture with automatically processable metadata descriptions of informal knowledge sources and completely formal knowledge parts in the control of the Information Agents (rule systems for post-processing and preconditions), shows a way into the direction of a combined / integrated handling of knowledge at different levels of formalization. This seems a pretty interesting and relevant research topic since informal knowledge representations are typically available in an affordable manner in the real-world, whereas formal approaches are required to realize powerful system functionalities.

KnowMore demonstration examples

We chose a presentation-in-context approach for describing our system functionalities which might appear pretty broad, but should enable the user to assess how practically *relevant* and practically *feasible* such services are. The examples were designed such that they should show the major, above-mentioned, features of the approach on realistic data, but also small enough to oversee and understand them and not to be overburdened with too many application details.

The **KnowMore purchasing example** should mainly demonstrate the idea of dynamic task context and pro-active, workflow-embedded information assistance. Further, at the hand of this example, we could imagine well the use of ontology-based background knowledge for improved Information Retrieval (if there is no business rule about buying graphics cards, there might be one about buying any PC card, ...).

The **KnowMore contact management** application should stress some advanced features, in particular business processes with loops as a control construct (several runs through the process model) and how this affects the dynamic task-context approach. Further, I could demonstrate the context-sensitive *storage* of information and the idea that stored process instances themselves represent first-order citizens of our information landscape (if there was already a contact to a certain company, the whole process instance with all detail could be found in the archive and analysed).

In general, the KnowMore implementation represents *one* operating point in the space of possible solutions spanned by our major design ideas. For concrete applications, some points may be more or less important. In some scenarios, specific elements of the overall, big picture may be more or less relevant and realistic. The goal of this thesis is not to promote one concrete operating point as the ultimate solution, but rather show the range of possibilities, discuss how those could be realized and let the potential user decide what parts of the picture to take and to realize in his / her own application environment.

3 A Conceptual Framework for Business-Process Oriented Knowledge Management

Nichts ist im Verstand,
was nicht vorher in den Sinnen war,
außer dem Verstand selbst.
G. W. Leibniz

Abstract: In this Chapter I go in more detail through the four layers of the already sketched KnowMore conceptual framework for a business-process oriented Organizational Memory Information System which realizes context-sensitive, proactive knowledge services. For all four layers, I discuss comprehensively the conceptual background and functional requirements, define them concisely, and characterize the range of possible realizations. Since I do not prescribe a fixed approach for implementing each layer, but rather analyse (1) basic definitions and requirements, (2) the relevant functional and architectural elements, as well as (3) possible realizations, the results of this Chapter essentially amounts to a reference architecture for a business-process oriented OMIS. The Chapter is structured as follows:

- I present the KnowMore Application Layer (AL) as a conceptual extension of the Workflow Management Coalition's (WfMC) reference architecture in Section 3.1.
- The KnowMore Knowledge Brokering Layer (KBL) hosts the data structures and functional elements for realizing intelligent knowledge services, tiggered by the AL and manipulating the KDL. The KBL is introduced in Section 3.2.
- The KnowMore Knowledge Description Layer (KDL) is the basic integrative element of our architecture, based upon rich, ontology-based metadata (Section 3.3).
- The KnowMore Knowledge Object Layer (KOL, Section 3.4) is constituted by the set of all relevant knowledge, data and information sources to be managed and exploited by the OMIS, plus possibly required wrapper modules for providing appropriate interfaces to the KDL.

- Finally, in Sections 3.6 and 3.7, I summarize the major contributions, its impact up to now, and relevant related work.

Preamble: *The abstraction of several earlier results into a generalized model, as well as the comprehensive and consistent discussion of the four layers with much detail has not been published before. Nevertheless, these results could not have been achieved without initial work done in the KnowMore project. Hence I have to mention Prof. Knut Hinkelmann as the first KnowMore project leader who was one of the major driving forces behind the idea of (re-)using workflow technology for the KnowMore application layer. Further I have to mention Ansgar Bernardi who elaborated the embedding of our approach into the Workflow Management Coalition's framework. The contributions of other colleagues, especially Michael Sintek, to the KnowMore project, have already been mentioned before. The idea of a systematic analysis of the application layer seen within the Grassroots framework, came from Ludger van Elst and was superficially described in [Elst & Abecker, 2001]. The work on elaborated context models for improved information delivery – as discussed in this Chapter – has initially been sketched in [Elst et al., 2001]. Earlier presentations of the context topic can be found in [Abecker et al., 2000c; Abecker et al., 2000e]. Heiko Maus took up the topic, combined it with earlier work in the areas of context exploitation for groupware and in document analysis (cp. [Wenzel & Maus, 2001; Maus, 2001]) and will hopefully accomplish his Ph.D. thesis about this topic.*

3.1 Application Layer

"... to make the knowledge so readily accessible that it can't be avoided.".

[Davenport, 2002]

3.1.1 Basic Motivation

One central motivation for all this thesis is to improve technologies which allow to provide a knowledge worker automatically with that information he or she actually needs for performing her task better.[27] Technically speaking, we need answers for two, interrelated questions:

- What information does the information system need in order to optimally fulfill an actual information need? This means, in which terms are queries formulated which go to the OMIS *(query constraints)* ?[28]
- How can this information be determined from the system, i.e. from which factors do the query constraints depend *(context factors)* ?

A simple illustration of this issue is given in Figure 21. What is finally needed in order to let the OMIS do its job, is shown at the right hand side which indicates the query to be answered at the end. The major influence factors are identified as task, role, and individual aspects or, context factors.

We do not discuss in depth User Modeling and Personal Profiling

The topic of personal influence factors is an advanced research topic investigated by a big research community (cf., e.g., [Kobsa, 2001; Setten, 2001]): considering, for instance, the effect of prior knowledge, personally specified interest profiles, or personal quer history in order to find out personal interest focus, personal rules for query disambiguation, or personal presentation preferences. Although it is obvious

[27] This is the local, short-term goal. On the other hand, we have global, long-term goals such as filling, improving and organizing the organizational knowledge base, foster the consistent use of best practices, harmonize organizational structures, behaviours, and taxonomy, etc.

[28] This is what [Steier, et al., 1995] calls *search constraints* like: from which information sources do we select information? What topic are we searching for? Are there any restrictions with respect to costs or maximum delay for an answer, ...

that for an industrial-strength, real-world implementation of our ideas, user modeling and personal profiling topics should definitely play an important role, this thesis will not discuss any details of this topic, since (i) it has not to contribute *new* ideas to these research issues, and (ii) it is clear how those ideas can be incorporated in the framework.[29] Furthermore, there was some good thread of work in this area undertaken in the CoMMA project [Gandon, 2003; Kiss & Quinqueton, 2001] which finally lead to the Ph.D. Thesis of Fabien Gandon [Gandon, 2002a].

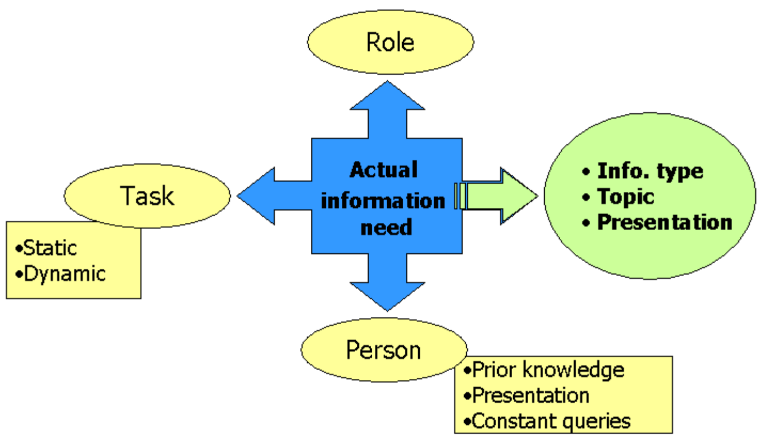


Figure 21: Context Factors of an Actual Information Need

Instead of these individual elements of information needs, let us focus on the organizational factors, i.e., on the question of organizational role and actual tasks as context factors for information needs. This topic is by far not yet a developed and mature research topic, and did almost not exist when the work on the topics of this thesis started.

Let us make another remark regarding task-specific influence on information needs. Here – as we saw already in the KnowMore application examples in Sections 2.2 and 2.3 and as we will see below in Subsection 3.1.2 – we have to introduce another refinement which is only indicated in Figure 21: Namely, that

[29] This will become more apparent later in this thesis, with the proposal of a new research topic on Agent-Mediated Knowledge Management.

task-specific influences can be further differentiated into local and global and into static and dynamic factors (which play together to some extent):

- **Static versus dynamic**
  - **Static** aspects: for some information needs, it might be clear at modelling (process analysis) time that always, in all possible instantiations of a given process, a certain task can be supported by a specific, already exactly determinable, information need.

    Static aspects influencing an actual information need

    For instance, when filling a given taxation form, it can be said completely in advance, that the official tax office regulations for filling this form are a relevant element of background knowledge.
  - **Dynamic** aspects: on the other hand, as seen in the KnowMore examples, there might be information needs which can not be fully formulated until process runtime because they are dependent on prior tasks or on environment factors.

    Dynamic aspects influencing an actual information need
- **Local versus global**
  - **Local** aspects: information needs which are only relevant for a given task and can be determined and answered exclusively with the information available at the level of this single task. This means, we do not need any input from other parts of the embedding process for sending a fully specified query to the OMIS.

    Local (to a task) influence factors
  - **Global** aspects: they characterize influences on a specific, situated information need which are not only dependent on the current task, but also from overall process characteristics, tasks executed before, or even tasks to be executed later.[30]

    Global (for a process) influence factors

[30] This idea goes beyond the scope of this thesis. Just to briefly sketch it: if we try to achieve an economically optimal modelling approach, we should aim at a complete separation of concerns in our modelling framework (person, role, task aspects, plus global and dynamic task context). The several influence factors could then be reused, changed, and maintained separately without many unexpected side-effects, and the knowledge could be combined in a most powerful manner. This approach is taken up to some extent in the idea of task ontologies which organize business tasks (to be used later as building blocks for process models) in task hierarchies which could also contain task-specific information needs (cp. [Schwarz, 2003; Fensel, 1998]). The concept of task ontologies is, by the way, not so far away from the MIT Process Handbook [Malone et al.,

For instance, the question which person initiated a given workflow (which is a global workflow parameter), might have an influence on the importance / priority of this process instance, and, hence, also on its tasks. Then, some information sources or delivery modes might be only relevant or applicable for very important or for less important process instances.

A detailed understanding of influence factors helps for designing analysis methods

This fine-grained differentiation might look too sophisticated for two reasons. One reason is that you might ask what it is good for. Here we can answer that – the better we understand all influence factors for the information needs in a given situation, the bigger is our chance not to oversee something relevant and potentially valuable when building an OMIS and modelling concrete information needs and support specifications. So, we have a methodological aspect.

Locality and static aspects are often interwoven, but not identical

The other source of criticism might come from the observation that "normally", local and static aspects correspond, as well as global and dynamic aspects. If this would be always the case, my distinction would not be minimal. It is right that in most cases this correspondence is really given – which is the reason that I used the terms local process context and task context, or global process context and dynamic task context, in the KnowMore examples above really often exchangeably or even synonymously. However, we can indeed imagine situations with:

Local influence of dynamics

- **Local, but dynamic task context influence:** assume an information need which is depending from the value of a context variable which is just set in this task itself, not earlier in the business process. Then, we have to wait until this task is being executed, before we can evaluate the information need, however, we don't have a dependency with another part of the overall process.

Static influence of global parameters

- **Global, but static process context influence:** we can assume cases where all steps in a process, i.e., all single tasks are influenced by the fact who initiated the process, as mentioned already above. Such overall process parameters as the process initiator can be considered rightly as static, because they are at least fixed and known at process start time.

2003]. In such a scenario, we can imagine easily that there might modifications be necessary; depending on the process context that a given task is reused in, the same job could be done pretty differently (depending on

Now having explained the basic idea of the three-dimensional influence space on situated information needs, before going deeper into the question of task- and process-specific information needs, let us have another short look on the appropriateness of exactly these three basic dimensions.

3.1.2 Task, Role, and Individual Aspects of Information Needs

Our description frame is oriented towards Winograd's Grassroots model

Our overview is based on and motivated by the "collection-mediated collaboration" model introduced by Winograd and colleagues with their Grassroots system [Kamiya et al., 1996]. Grassroots unifies personal and organizational information management by integrating several modes of information transfer that are – currently – provided by independent tools in the most cases. These information transfer modes are characterized by three dimensions:

Dimensions to describe information transfer modes

- the **regularity** (continuous vs. ad-hoc),
- the **initiator** (information provider vs. consumer), and
- the **delivery** (with vs. without notification)

of an information transfer. These notions allow, for instance, a clear definition of push (information transfer initiated by the source) and pull services (information transfer initiated by the destination).

Ordered according to the predominant information-need determinant, we describe below the main "use cases" for intra-organizational information transfer and show the respective roles of the several profile parts introduced above. You may recognize elements of the KnowMore sample applications in these examples, too.

3.1.2.1 Task-Driven Information Services

- **Process-Embedded, Ad-Hoc Information Pull:** A workflow management system (WfMS) interprets the process logic represented in a business process model. It assigns each task to appropriate actors by matching the possible roles of the task with the roles of the users. When a task is processed, the generic task information need is instantiated (e.g. with information from previous activities via flow variables) and modified by the role and user profiles. For example, consider a task "search for literature" in a research project. The task-oriented portion of the information need comes from the project's domain (e.g.

what exactly is done before or after).

"agent technology in KM"). The task itself should be cooperatively solved by the project leader, a researcher, and an assistant. Clearly, each of these roles has different information needs. While the project leader might focus on other finished projects within the project's domain, the researcher would be interested in the latest journal and workshop articles, and the assistant - responsible for the implementation - might need detailed information about agent platforms. The individual employee who performs the task, additionally influences the actual information need with his experience and knowledge. Information about a task-related topic where he has much knowledge might be ranked lower; rookies get more basic information, and experts more details, etc. Moreover, individual presentation preferences (e.g. language) can be considered. Except for the personal profile aspects – which were not explicitly addressed in that project – this is the typical KnowMore case as described in Section 2.

- **Process-Embedded Information Prefetch:** Process enactment by a WfMS offers knowledge about presumable future tasks. Thus, instead of waiting until a task is actually executed, information about a forthcoming task and the role of its potential processor[31] can be used for information prefetch. This is useful when time consuming, difficult information searches must be performed, e.g., consulting and integrating many sources – maybe outside of the own organization, or – in a globally acting organization – accessing personal knowledge of people working in another time zone. At the moment when an actual user can be assigned to the task, the result of the information prefetch is at his or her disposal and may then be individualized (e.g., filtered or re-ranked according to the user's profile).

- **Task-Oriented Ad-Hoc Information Pull:** Often, highly knowledge-intensive processes are not formally modeled, because they are too complex or because they are too much ad-hoc (see also Section 5.3). *Attentive systems*, e.g. personal information agents like Watson [Budzik & Hammond, 1999], try to detect the task a user is actually performing, and use this knowledge to retrieve context-oriented information. In contrast to the process-embedded scenarios above, only the *local* work context can be obtained here (e.g., the application

[31] This means, the actual processor may be unknown at that time.

program that a user is working with). Thus, relevant knowledge from preceding tasks is hardly available for better specifying the actual information need. However, the integrated modeling approach allows for an exploitation of the user's roles for more precise assumptions about his or her current task.

3.1.2.2 Role-Oriented Information Distribution

- **Continuous, Role-Oriented Information Distribution:** Often, some specific information elements have to be distributed to all users that fill a specific organizational role. For example, an up-to-date version of the guideline for preparing project reports should be sent to all project leaders. This is modeled in the role's information need. The role "project leader" can be used as a kind of intensional description of a group of addressees. This description is expanded to generate the extension (e.g., an e-mail list) by resolving the role-user associations. Knowledge about the current task of a recipient then helps to determine the actual presentation strategy. Users that are currently involved in an *important* other task, do not get a high-priority notification, while project leaders that are just preparing a report may be interrupted in the case that their actual work is affected. Due to the separation of intensional information need descriptions and addressing schemas and the extensional, explicit storage of actual enactors of each role, the information push is easier to maintain. It is also more user-friendly, because of the task-orientated presentation.

- **Ad-Hoc Notification of User Groups:** Additionally, intensional specifications of addressees can be used for an ad-hoc information push with direct notification: E.g. "all secretaries that are preparing an invoice must recognize that the VAT generated by invoice software is no longer valid; thus it must be changed by hand". In this "alarm scenario", the task and the role model are solely used to state who has to get some information.

3.1.2.3 User-Oriented Information Services

- **Effective Personal E-Mail Management:** In personal e-mail, the recipient's information need is (or, should be) presumed by the sender. Knowledge about the roles and tasks of a user can be utilized to adequately process the mail. For example, the message can be put into a role- or task-specific mail folder, and the notification mode can be adjusted, depending on the relevance for the task at hand.

- **Interest-Based, Continuous Information Pull:** A continuous information pull which is based on a user's personal interests (this is the usual case in User Profiling scenarios), can be refined by his or her potential organizational roles. For instance, in a subscribed newsgroup for a specific software system, the purchasing agent in a company will mainly be interested in information about new products and prices, while a system administrator is interested rather in installation procedures and troubleshooting.

3.1.2.4 Conclusions

Since a major objective of this subsection is to show that task and role aspects may provide valuable input for improved information services, we first categorize the examples given according to the Grassroots dimensions. If we see that those examples cover a significant part of the design space spanned by the Grassroots dimensions, and if, further, in all – or most of – these examples it can be seen that task and role aspects could have a significant influence on information needs, this should show that it makes sense to have a closer look at these aspects. So, let us first consider the classification given in Table 13.

	Regularity of information transfer	*Initiator of information transfer*	*Delivery mode*
Process-Embedded Ad-Hoc Information Pull	ad-hoc	system for information consumer	with notification[32]
Process-Embedded Information Prefetch	ad-hoc	system	n/a
Task-Oriented Ad-Hoc Information Pull	ad-hoc	system for information	with notification[33]

[32] Of course, one could argue that the delivery mode as shown earlier in the KnowMore system, is without notification. But, one has to see that the information button "I" is nothing else than a maximally unintrusive delivery notification which indicates that "there is something", independent from the question whether you want / will to have a look at it. But there must be kind of active notification, since the user does not necessarily know that there is some information offer.

[33] Same remark as above: Systems like Watson exactly have been *built* to provide proactive hints to potentially interesting information.

		consumer	
Continuous, Role-Oriented Information Distribution	continuous	information provider	both modes possible (typically without notification)
Ad-Hoc Notification of User Groups	ad-hoc	information provider	with notification
Effective Personal E-Mail Management	ad-hoc	information provider	both modes possible (typically with notification)
Interest-Based, Continuous Information Pull	continuous	information consumer	both modes possible (typically without)

Table 13: Classification of Given Examples wrt. Grassroots Dimensions

Remarks from analysing the OMIS scenarios

These seven different sample scenarios should together provide a good, representative sample set of possible application modes of information systems. Some remarkable observations can be made when analysing the scenario overview of Table 13 one steep deeper:

In an OMIS scenario, a new transfer initiator occurs, the system itself

1. The Grassroot classification had to be extended in one respect: in the upper three examples, the transfer initiator is neither the information consumer nor the information provider. Instead, the OMIS occurs as a mediator which acts on behalf of the user. In principle, this is a consumer-oriented scenario, but nevertheless it should be noted that here some new quality of system services arises.

The OMIS may even act on its own behalf

2. The second scenario (Process-Embedded Information Prefetch) is in this respect even more unusual: Here, the system acts more or less on its own behalf in order to improve efficiency of services. Of course, the principle operation mode is consumer-driven, and the end user has an indirect benefit. But in the first instance, the direct beneficiary is the system itself.
   - This causes another “irregularity” of this scenario: the fact that the category “notification mode” is not really applicable. The reason is that at the time when this information transfer is performed, the consumer of the information is the system and not a human user.

In an OMIS, the notification mode is often a matter of personal preferences

3. Maybe the most interesting observation is that a number of scenarios can be run reasonably in both notification modes, and that – in a concrete situation – the decision which mode to use can be derived (dynamically,

depending on actual situative data, as well as statically, based on a fixed information consumption profile) from task, role, or personal information need profiles.

We summarize the importance of influence dimensions for the sample scenarios

A somehow informal, but – hopefully – nevertheless comprehensible summary of the examples of the subsections above, is given in Table 14. Here, the influence of the several determinants of concrete, situational information needs is assessed, as presented in the examples above. The marks are read as follows:

- xxx: predominant influence factor, which mainly constitutes/causes the information need described in the situation
- xx: influence factor with high importance
- x: less important influence factor which can be used to refine / improve given results of information search
- (x): influence factor which could be used for refinement / improvement if available, but not described, assumed, or typically available in practice, when considering the concrete example situation

	Task influence	*Role influence*	*Personal influence*
Process-Embedded Ad-Hoc Information Pull	xxx	x	x
Process-Embedded Information Prefetch	xxx	x	
Task-Oriented Ad-Hoc Information Pull	xxx	(x)	(x)
Continuous, Role-Oriented Information Distribution		xxx	(x)
Ad-Hoc Notification of User Groups	xx	xxx	
Effective Personal E-Mail Management		x	xxx
Interest-Based, Continuous Information Pull		x	xxx

Table 14: Information Need Determinants in the Examples Above

Although one might argue about one or the other specific statement in this table, it should nevertheless be undisputed that:

- for each influence factor, there are use cases which are predominantly determined by it, and others where this factors plays (almost) no role
- in all use cases, more than one influence factor is relevant, in many cases, all three influences play together

So, even if it is difficult (at least, in practical implementations) to claim a clear orthogonality and independency between these dimensions, it should nevertheless be acceptable that for a thorough theoretical analysis, these factors should be considered separately.[34] Even if in all scenarios the actual information need is an amalgamation from task, role and user aspects, a discrete modeling would allow for better reuse of single models, for a better maintenance and easier changes, and for a more flexible utilization.

In practical scenarios, all three dimensions influence an actual information need

To sum up at the end of this parenthesis subsection, let us agree that it makes sense to care about a task- and role oriented architecture for the application layer of our conceptual framework. Hence the next step would be to gather some basic definitions for talking unambigously about these topics.

Hence software architectures should provide for a clear separation of concerns in this respect

After these basic considerations, let us find a clear conceptual basis for defining information needs.

3.1.3 Conceptual Foundations

First, for organizing the Application Layer definitions, let us introduce the four conceptual areas of the Core Enterprise Ontology (CEO) proposed by [Bertolazzi et al., 2001] on the basis of an analysis of the most prominent at that time existing Enterprise Ontologies, namely (1) IDEF5[35] from the Computer-Aided Manu-

Toward a Core Enterprise Ontology (CEO)

---

[34] Since this thesis proposes just a specific kind of innovative information system, but not a comprehensive, overall, organizational IT infrastructure plus modelling methodology, I do not fully elaborate this idea in all consequences. As already said, we neglect the personal profiling dimensions; and we do not keep task and role elements completely separate, but instead model tasks at a more fine-granular level such that they are to be enacted by exactly one organizational role. Thus, such a task-specific information need reflects task and role aspects together. Nevertheless, we consider a clear separation of concerns as interesting for future work. Some aspects have already been elucidated by [Elst & Schmalhofer, 1999; Schmalhofer & Elst, 1999]. Such a clear analysis of organizational information needs could be well reflected in an Agent-Mediated KM scenario as suggested in Section 5.2.

[35] http://www.idef.com/

facturing area; (2) PIF – the Process Interchange Format from MIT[36]; (3) the Business Engineering Model BEM[37] established in the UML world; (4) the Enterprise Ontology of the AI Applications Institute AIAI in Edinburgh [Uschold et al., 1998][38]; (5) the TOVE Ontology developed in the Toronto Virtual Enterprise project [Fox & Gruninger, 1998][39], and, finally, (6) the MIT Process Handbook [Malone et al., 1999; Malone et al., 2003][40].These four conceptual areas are described as follows (with slightest modifications):

The four conceptual main areas of CEO concepts

- **Transformations:** are enabled by active entities, they produce, consume, or access passive entities, and they represent arbitrary actions or processes in the organization.
- **Active Entities:** represent active elements in an enterprise, making decisions and performing actions.
- **Passive Entities:** represent business objects, i.e. passive elements to be created, accessed, modified, etc.
- **Conditionals:** represent expressions which can be tested for being satisfied or not, and used for describing business goals or for specifying preconditions of transformations.

The definitions extend existing work

In the following we will introduce the basic notions and definitions required for a clear understanding of the Application Layer of our generic OMIS framework. Naturally, this is more or less a conservative extension of notions and definitions already existing in the areas of Enterprise Ontologies[41], Business Process Management, and Workflow Management. Hence we have to recapitulate some material which is not original work contributed by this thesis, but is necessary to know for having a complete picture. We try to keep the repetition of existing work

[36] http://ccs.mit.edu/pif/

[37] http://www.mdcinfo.com/OIM/models/BRM.html

[38] http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html

[39] http://www.eil.utoronto.ca/tove/ontoTOC.htm

[40] http://ccs.mit.edu/

[41] The term "Enterprise Ontology" is established in the literature, even if it covers many concepts which are not exlusively interesting in the commercial world. So, we keep the term Enterprise Ontology to refer to this entity, but often make slight adaptations in the wording in order to show that also non-commercial organizations, like governmental institutions, are covered. In principle, an Enterprise Ontology happens to be a proper superset of a general Organizational Ontology, at least for the purposes of this thesis. Hence we will understand both terms synonymously in this context.

as short as possible and try to point out where existing definitions were reused and where extensions or changes were required, respectively.

The semi-formal notation follows [Uschold et al., 1998]'s notation.

Regarding the notation, the AIAI's approach of informal / semi-formal natural-language definitions for communicating ideas and clarifications is employed [Uschold et al., 1998]. Many of these concepts are also implemented in a *formal* ontology, but presumably the informal presentation is more appropriate for the purpose of this thesis. Like [Uschold et al., 1998], words and concepts in CAPITALIZED LETTERs represent formally defined concepts. It is useful to make explicit the distinction between these "technical terms" and the use of words in a common-sense, non-technical meaning. If some DEFINED CONCEPT is used in the following, but not explained in this thesis, then it is supposed that its detailed definition is not urgently required for understanding our argumentation line in this thesis. The respective definitions can be found in [Uschold et al., 1998] or in other, explicitly cited literature.

Finding a common Core Enterprise Ontology is still work in progress

As [Partridge & Stefanova, 2001; Partridge, 2002] point out, a generally agreed and applicable Enterprise Ontology does not yet exist. There are bits and pieces which can be critized in all existing partial approaches. Hence we had to make small changes and adaptations, and, often, elements from the AIAI Enterprise Ontology and from the Workflow Management Coalition's reference model and terminology were merged. Sometimes we will mention open or unclear points in this merging process, for discussion and further work. However, this does not affect the viability and reliability of the definitions in this thesis, since the existing work mainly played the role of a "host system" where new ideas and extensions were implanted. In the case of changes, the extensions should be applicable without too much work to be redone. Further, the fact that we do not discuss in detail a full formalization of the definitions presented, should not reduce the usefulness of the argumentation too much, since the major objective is to make clear the basic ideas, still abstracting from concrete implementations, and not a direct implementation of the Enterprise Ontologies for some formal, automated inferences.

Let us begin with some fundamental notions – mainly taken from the AIAI Enterprise Ontology – before we come to process- and organization-specific definitions.

An **ENTITY** is a fundamental thing in the domain being modelled.

- An ENTITY may participate in RELATIONSHIPS with other ENTITIES.

A **RELATIONSHIP** is the way that two or more ENTITIES can be associated with each other.

- A RELATIONSHIP itself is an ENTITY.

An **ATTRIBUTE** is a RELATIONSHIP between two ENTITIES (called the "attributed" and the "value" ENTITY) with the following property:

Within the scope of interest of the model, for any particular ENTITY the RELATIONSHIP may only exist with *only one value* ENTITY.

A **RELATIONAL ROLE (RR)** is the way in which an ENTITY participates in a RELATIONSHIP.

- Technically, when representing an n-ary RELATIONSHIP mathematically as an n-tuple, each possible RR associated with this RELATIONSHIP can be mapped to one specific position in this tuple.

An **ACTOR ROLE** is a kind of RELATIONAL ROLE (RR) in a RELATIONSHIP where the playing of the RR entails doing or cognition.

Like [Uschold et al., 1998], we use the word ENTITY sometimes for a *type* of ENTITY (also called a *class*) and sometimes for a *particular* ENTITY of a certain type (frequently called an *instance*). It should be possible to distinguish the two meanings within a given context. In the mathematical sense, an ATTRIBUTE is a functional RELATIONSHIP.

3.1.3.1 Definition of Active Entities

Here, we only mention the active entities required later for defining OMIS-relevant concepts. More details can be found in [Bertalozzi et al., 2001] or other Enterprise Ontology proposals. Let us begin with the most fundamental concepts, directly taken from [Uschold et al., 1998][42]:

Fundamental active entities

A **PERSON** is a human being.

A **MACHINE** is a non-human ENTITY which has the capacity to carry out functions and / or play various roles in an organization.

[42] With the small change that we include ORGANIZATIONAL ROLE and ORGANIZATIONAL POSITION as POTENTIAL ACTORs, two concepts which are not defined in the AIAI ontology.

An **AGENT** is a PERSON or a MACHINE.

For a particular point or period of time, an **ACTOR** is an ENTITY that actually plays an ACTOR ROLE in a RELATIONSHIP.

A **POTENTIAL ACTOR** is an ENTITY that can play an ACTOR ROLE in a RELATIONSHIP, i.e. it is an ENTITY for which some notion of doing or cognition is possible. The set of POTENTIAL ACTORs includes: PERSONs, ORGANIZATIONAL ROLEs, ORGANIZATIONAL UNITs, ORGANIZATION POSITIONs, and MACHINEs.

The notion of POTENTIAL ACTORs corresponds to the WfMC notion of a Workflow Participant and is used in this thesis synonymously.

Slightly changing the definitions of the AIAI Enterprise Ontology, we can define:[43]

An **ORGANISATIONAL UNIT (OU)** is an ENTITY (with a defined identity) for MANAGING the performance of ACTIVITIES in order to ACHIEVE one or more PURPOSES. An OU may be characterised by:

- the nature of its PURPOSE(S);
- one or more PERSONS working for the OU;
- RESOURCES allocated to the OU;
- other OUs that MANAGE or are MANAGED_BY the OU;
- a set of ORGANIZATIONAL ROLES associated with this OU;
- its ASSETS;
- its STAKEHOLDERS;
- being LEGALLY OWNED by an ORGANIZATION;
- its MARKET (if it is a VENDOR).

Please note that via the MANAGE and MANAGED_BY links, sort of a tree or directed acyclic graph structure between OUs can be built up which does not necessarily correspond directly to a set inclusion between the groups of PERSONs working for the affected OUs.

[43] In detail, our proposal is to add that an OU is LEGALLY_OWNED by an ORGANIZATION – with the suggestion to replace the original Enterprise Ontology concept CORPORATION by ORGANIZATION as it is foreseen in [Uschold et al., 1998]. We propose to define an ORGANIZATION – which is, together with PERSONs and PARTNERSHIPs – a LEGAL_ENTITY as (1) a group of PERSONs recognised in law as having existence, rights, and duties distinct from those of the individual PERSONs who from time to time comprise the group; and (2) being an OU which is not MANAGED_BY some other OU. This means, an ORGANIZATION is the root of the tree or directed acyclic graph spanned by the MANAGES relationship.

The notion of ASSETs will be extended by KNOWLEDGE ASSETs to be defined later in this thesis (Section 3.4).

Since we introduced the notion of an ORGANIZATION (see above), we had also to change the LEGALLY OWNED clause.

The notion of organizational roles was added to the AIAI ontology

The most important difference to the original AIAI definition is probably that we explicitly mention ORGANIZATIONAL ROLES thus linking into an explicit *organization model* in the sense of typical process modelling languages in Business Process Modeling [Junginger et al., 2000; Scheer, 2001; Böhm & Schulze, 1995] or Software Process Modeling [Acuna & Ferré, 2001; Finkelstein et al., 1994; Rombach, 1988;]. The idea of bundling rights and responsibilities into a formal role concept associated with people or positions in an OU, is not expressed that explicitly in the AIAI ontology. There are some interesting approaches for modeling organizational roles. A relatively comprehensive proposal has been made by [Fox et al., 1995] in the TOVE project characterising an organisational role by goals, required skills, associated processes, policies, and information-links. Information-links for describing communication between organizational agents are an interesting approach here, since they show the direction towards an *organizational communication anaylsis* – a promising idea with respect to knowledge-process optimization (cp. [Remus, 2002; Dämmig et al., 2002]). However, for the purpose of this thesis, a relatively lean definition of an organizational role is already sufficient. Hence we can add the following:

Roles and positions together with organisational units, constitute the organization structure

An **ORGANIZATIONAL ROLE (OR)** can be played over some period of time by a PERSON or – theoretically – a MACHINE, i.e. by an AGENT within an ORGANIZATION.

The OR is either defined in the context of one or more permanent (like a department) or a temporary (like a project team) ORGANIZATIONAL UNITs or within the scope of one or more PROCESSes or PROCESS INSTANCEs.

The OR is characterized by a set of rights and obligations with respect to this defined scope, which technically means by a set of ACTIVITIES that the AGENT who plays the OR, must perform or is allowed to perform.

An OR might be associated with a set of POLICIES which define constraints on the way how to perform the respective ACTIVITIES (e.g. with respect to quality, resource consumption, etc.).

An **ORGANIZATIONAL POSITION (OP)** defines a formal position within an OU that can be filled over a period of time by a PERSON.

An OP essentially consists of a set of ORGANIZATIONAL ROLEs which have to be carried out by the PERSON filling the OP.

An OP might further be characterized by a set of POLICIES, i.e. constraints on the way how to perform ACTIVITIES and how to enact PROCESSES when filling the associated ORs. POLICIES are inherited from the associated ORs.

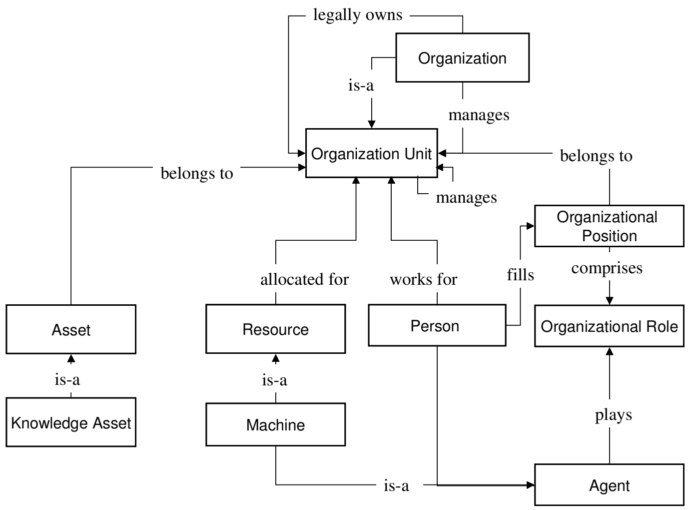


Figure 22: Some Central Concepts Regarding Active Entities

An individual agent can assume several ORGANIZATIONAL ROLEs at the same time. Vice versa, one ORGANIZATIONAL ROLE might be played at the same time by different ACTORS. Examples for ORGANIZATIONAL ROLEs include "project manager", "code reviewer", "IT budget manager". Examples for ORGANIZATION POSITIONs are "President of Corporation", "Member of the Board", "Senior researcher" (cp. [Fox et al., 1995] and Figure 22)

3.1.3.2 Definition of Transformations

The central concepts in the realm of transformations are all around the notion of a Business Process. In the following, we introduce the basic concepts in this area. The definitions are mostly merged from the Workflow Management Coalition's [WfMC, 1999] view and the AIAI Enterprise Ontology, with some terminological adjustments.

Activities are the basic building blocks of transformations

An **ACTIVITY** is something done or to do over a particular TIME INTERVAL, representing a piece of work that forms one logical step within a PROCESS. The following may pertain to an ACTIVITY:

- has PRE-CONDITIONs

- has EFFECTs
- is performed by an ACTOR
- entails use and/or consumption of RESOURCEs
- has AUTHORITY requirements
- is associated with an (ACTIVITY) OWNER
- has a measured efficiency

A **MANUAL ACTIVITY** is an ACTIVITY which is done or to do by an ACTOR which is a PERSON.

An **AUTOMATED ACTIVITY** is an ACTIVITY which is done or to do by an ACTOR which is a MACHINE.

A **KNOWLEDGE TASK** or **KNOWLEDGE SERVICE** is an ACTIVITY which performs a MNEMONIC FUNCTION.[44]

The word "task"is often used as a synonym for an ACTIVITY.

Activities as the smallest scheduled unit of work

In contrast to the AIAI definition, we adhere to the WfMC view, thus not allowing the decomposition of ACTIVITIES into SUB-ACTIVITIES, but rather understanding an ACTIVITY as the smallest unit of work scheduled by some enactment machinery at process runtime. This is for the sake of terminological compliance with the WfMC-standardized metamodel and is sufficient for "normal" business processes and workflows to enact them. The necessary element of hierarchical decomposition comes into play when defining decomposable PROCESSes and PROCESS SPECIFICATIONs.[45]

For the crucial notions of PRE-CONDITIONs, EFFECTs, ACTORs, and RESOURCEs see the Subsections on Conditionals, Active Entitites, and Passive Entities, respectively.

Knowledge tasks as the OMIS specific extension of existing concepts

The notion of MNEMONIC FUNCTION is specific to an OMIS, goes back to

[44] Informally, a MNEMONIC FUNCTION is each act which creates, stores, manipulates, or retrieves a part of the OMIS, i.e. in particular knowledge content stored in the OMIS. A bit more formal, this means that the EFFECT of a KM TASK must be formulated such that it affects some part of the OMIS. For a more detailed discussion of MNEMONIC FUNCTIONS, see Section 3.2 on the Knowledge Broker Layer.

[45] Although we use the WfMCs interpretation, the AIAI approach is also convincing from the perspective of "ontological clarity". If one would go deeper into the idea of weakly-structured workflows composed from on-the-fly configurable and decomposable ACTIVITY SPECIFICATIONs that are arranged in some task (or, activity) hierarchies (as motivated in Section 5.3), it could make sense to revise this design decision in the definitional framework and come back to the AIAI approach. But, ideally, this would be done collaboratively with the research community working on flexible and ad-hoc workflows.

seminal work on Organizational Memory, and has been refined in the context of Information Technology by [Klamma, 2000]. Obviously, also the notion of a KNOWLEDGE TASK is new. We will use the terms KNOWLEDGE TASK, KM TASK, and KNOWLEDGE SERVICE as synonyms.

Mainly following [Remus, 2002] we characterize a KNOWLEDGE-INTENSIVE ACTIVITY as follows:

Knowledge-intensive activities are a special case of activities

A **KNOWLEDGE-INTENSIVE ACTIVITY** or, a **KNOWLEDGE-INTENSIVE TASK (KIT)** is an ACTIVITY which:

- is typically a problem-solving, decision, judgment, or management task;
- often exhibits the properties of a “wicked problem” or a “fuzzy task”, according to [BuckinghamShum, 1997; Zigurs & Buckland, 1998; Conklin & Weil, 1997];
- tends to be much communication-oriented, information-processing, and / or argumentation-based;
- differs much in enactment quality and efficiency when performed by different people, especially depending on the human’s prior knowledge and experience;
- has (among other things) the EFFECT of changing the values of DECISION VARIABLEs;
- may be facilitated by a (set of) KNOWLEDGE SERVICE(s).

We see that this definition is not “crisp”, but also, to some extent, fuzzy – which is not surprising in this area.[46] For the definitions of EFFECT and DECISION VARIABLE’s please refer to the following Subsections discussing Conditionals, and Passive Entities, respectively. Please keep in mind that KNOWLEDGE SERVICE is a synonym for KM TASK.

Further characteristics of knowledge-intensive activities

KITs are normally enacted by the human user, not by software agents. There is an exception: AI-based Expert Systems: tasks which could only be automated as an Expert System, are definitely KITs. However, a major goal of this thesis is just to replace the necessity for expensive and difficult to maintain Expert Systems, by more lightweight and human-oriented Assistant Systems. Hence, the observation is not really useful.

We may also see that KITs are mostly not well-suited to be further decomposed for finding a well-structured, workflow-oriented support. Rather, they are the level of

[46] For a more detailed discussion of characteristics of KITs, please refer to Subsection 4.3.

granularity where further task refinement becomes difficult, and thus they are treated by the workflow management system by one single task. Here, the kind of system support switches from coordination support to information and knowledge supply, which is then task and domain knowledge, instead of process knowledge.[47]

Now that we have analysed the concept of ACTIVITIES, we can compose BUSINESS PROCESSes from them. BUSINESS PROCESSes stand at the core of our considerations, since they are the ultimate goal of our optimization endeavour. Therefor, we combine and adapt the definition of the Workflow Management Coalition [WfMC, 1999] and elements / concepts of the AIAI Enterprise Ontology [Uschold et al., 1998]:[48]

Business processes stand at the core of our interest

A **BUSINESS PROCESS (BP)** is a set of one or more linked ACTIVITIES which collectively HELP ACHIEVE a PURPOSE within an ORGANIZATION, or in a collaboration between several ORGANIZATIONs.

- "Linked" means that normally there holds a number of temporal and logical relationships (CONDITIONS) between ACTIVITIES which together induce an implicit set of rules and regulations (call it: business logic) which govern the exact running of a BP.
- A BP is executed by a set of ACTORS who perform specific parts of the BP – i.e. specific ACTIVITIES contained therein – according to the ORGANIZATIONAL ROLEs and POSITIONs they fill.
- It uses or consumes a set of RESOURCES, it has tangible and intangible input, and normally produces a (set of) PRODUCT(s) and / or SERVICE(s) as OUTPUT, thus involving some creation of value-added for the ORGANIZATION.
- As each PROCESS, a BP performs some transformation or transport of matter, information, or energy from a defined start to a defined final state, following determined rules.
- Within the limits of the induced business logic (see above), a BP is typically executed many times in a similar manner, for dealing with different business cases.

[47] In German, this is the step switching from "Prozesswissen" to "Funktionswissen", as it is called, e.g., in the ARIS methodology [Scheer, 2001]. In our opinion, this is more a smooth transition than a crisp separation (the only hard distinction criterion is the matter of atomicity of a task / activity / function which should be done by one person as one logical working step. But in the case of knowledge work, we may even imagine cases where such an atomar, logical working step is done by a collaborating team where only the final result is brought out of the group, but the way of working on the topic is transparent for the overall system). To reflect such sophisticated considerations with respect to process support, is an aim of the weak-workflow topic to be introduced later, in Section 5.3.

[48] See also [Stahlknecht & Hasenkamp, 2002] and [DIN 66201].

> • Often, a BP can also be considered as composed from sub-processes which consist of a number of involved ACTIVITIES in such a way that they can be seen themselves as BPs.

Real-world activities are modelled through activity specifications

Now we proceed from the level of "real-world" entities to the "model world" represented in the computer. For the basic concept of an ACTIVITY SPECIFICATION, we start from AIAI's definition and extend it in order to make the necessary provisions for allowing pro-active, context-sensitive knowledge services.

Activity specifications are the elementary components of process models

> An **ACTIVITY SPECIFICATION** is a characterisation of something to do, i.e. a specification of an ACTIVITY, using a formal specification language.
>
> Hence it may contain (in an explicit or an implicit manner) unambigous characterisations of all possible properties of the respective ACTIVITY, i.e. in particular, PRE-CONDITIONs, EFFECTs, ACTOR, RESOURCEs, AUTHORITY requirements, and OWNER.

Since we use the words activity, task, and function synonymously, an ACTIVITY SPECIFICATION can also be called a TASK MODEL or a FUNCTION SPECIFICATION.

As ACTIVITIES were specialized into KNOWLEDGE-INTENSIVE ACTIVITIES, it is not surprising that this is also reflected at the modelling level:

KIT SPECIFICATIONS describe knowledge-intensive workflow tasks.

> A **KIT SPECIFICATION** is the ACTIVITY SPECIFICATION of a KNOWLEDGE-INTENSIVE ACTIVITY. In addition to the properties inherited from the definition of an ACTIVITY SPECIFICATION, the following holds:
>
> • It may have (inter alia) the EFFECT of changing the values of DECISION VARIABLEs.
>
> • It may be characterized by a SUPPORT REQUEST to describe potentially useful, supporting KNOWLEDGE SERVICEs.

For the definitions of EFFECT and DECISION VARIABLE, see below the Subsections on Conditions and Passive Entities, respectively. A detailed example for a concrete KIT SPECIFICATION and the underlying formal language has already been shown earlier, in Table 9 and Table 10. There, an EFFECT is not stated explicitly. Coming to the particularities of Knowledge-Intensive Activities:

> A **SUPPORT REQUEST** (SR) – usually belonging to a KIT SPECIFICATION – is an ACTIVITY SPECIFICATION for an AUTOMATED ACTIVITY S which may start, run, coordinate, and further process the results of a (number of) KNOWLEDGE SERVICE(s) in order to HELP ACHIEVE that the ACTOR of the associated KIT – if using the results of the KNOWLEDGE SERVICE execution – will perform the associated KIT in such a way that the related INTENDED PURPOSE will be reached better.

Further, the following conditions shall hold for the SUPPORT REQUEST:

- If SR is linked to a KIT SPECIFICATION K, then its set of PRE-CONDITIONS contains at least the set of K's PRE-CONDITIONS, plus an additional constraint that K must have been started by an ACTOR in order to start also the support activity S
- SR encompasses a number of INFORMATION NEEDS which in turn CONTRIBUTE to a (possibly empty) set of DECISION VARIABLES
- SR may access a (possibly empty) set of CONTEXT VARIABLES
- SR may contain a POST-PROCESSING SPECIFICATION

To clarify the relationships a bit, have a look at Figure 23.

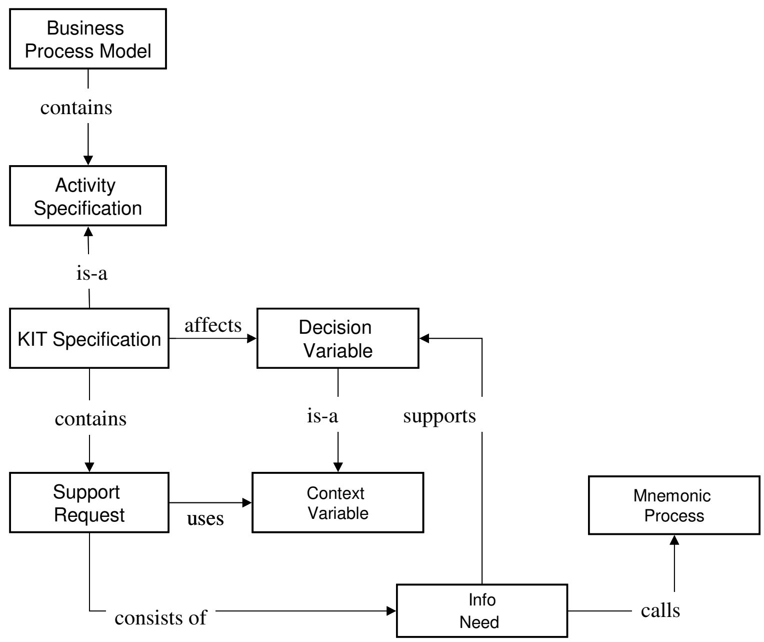


Figure 23: Central Concepts in the Area of Knowledge-Intensive Tasks

For these central definitions, a number of explanations is appropriate:

Support and operational activity shall usually start together

- The fact that the support specification's pre-conditions are a subset of the associated activity specification's pre-conditions technically means that the support activity S can be started whenever the associated activity can be started, depending on possibly available, additional conditions which may refer, e.g., to the overall process status. The fact that an additional condition is inserted referring to the start of the associated activity, ensures that the automatic support activity is activated simultaneously, not earlier. In the case that we would have some information prefetch mechanism (as mentioned in the introductory part of this Chapter) – which is today not usual, it might make sense to relax the definition in this respect such that a supporting activity can even start well before the supported activity is running.

Let us define an **INFORMATION NEED** as a pair consisting of a (set of) DECISION VARIABLE(s) and a MNEMONIC PROCESS.

Further, a **POST-PROCESSING SPECIFICATION** is a formal procedure which describes how to analyse, integrate, combine, transform and format sets of KNOWLEDGE ITEM DESCRIPTIONS[49] as provided by evaluated INFORMATION NEEDs in such a way that they can be either (a) provided to the end user through the information browser in the case of retrieval functionality; or (b) handed over to the KBL and KDL for storage in the case of a knowledge acquisition functionality; or (c) used internally in the KBL for performance optimization in the case of a learning functionality.[50]

- In the case that no POST-PROCESSING SPECIFICATION is given in a SUPPORT REQUEST, this would imply that no INFORMATION NEEDS may perform a retrieval functionality, since every KNOWLEDGE ITEM DESCRIPTION must be "stripped" before it can be sent to the information browser, because the end user is normally only interested in the content, not in all metadata. So, there will hardly be a presentation without a post-processing.

What is the meaning of an empty post-processing specification?

- The set of **DECISION VARIABLEs** influences the way that the KM support services are offered to the user: the normal case is that an INFORMATION NEED starts an information retrieval process which will hopefully find a number of potentially useful material in the OMIS knowledge base. This

Decision variables specify the exact target for information delivery

49 For a formal definition of KNOWLEDGE ITEM DESCRIPTIONs (KIDs), please refer to Section 3.3 about the Knowledge DescriptionLayer. For the moment it should be sufficient to remind the examples from the KnowMore Chapter above for an intuitive understanding, and to think of them technically just as DATA OBJECTs in the sense of the Subsection on Passive Entities.

50 Here we presume already a basic understanding of the structure of possible MNEMONIC FUNCTIONS which are explained later in the Subsection 3.2 on the KBL.

material is then presented to the user in an appropriate manner, normally through the information browser. As we saw in the KnowMore examples, the link between INFORMATION NEED and DECISION VARIABLE helps here to present found material at the right time in the right place. It also helps to re-evaluate an INFORMATION NEED when relevant conditions change.

What is the meaning of a support specification without related decision variables?

- However, it is also possible that the specified set of DECISION VARIABLES is empty (for the whole SUPPORT REQUEST, but for a specific INFORMATION NEED as well). This is possible in two cases: (1) we have also a retrieval functionality, but for any reason the user was not able or willing to make an exact assignment of info needs to variables. Then all possible support offers produced by the INFORMATION NEED(s) evaluation must be understood as targeting at the associated ACTIVITY as a whole. Hence the material will be presented unspecifically as an unstructured set. (2) We don't have a retrieval functionality, but some other MNEMONIC FUNCTION which is not associated to a DECISION VARIABLE (like storage of information).

What is the meaning of an empty set of context variables?

- The set of **CONTEXT VARIABLES** that SR is associated with determines the data which is accessible for contextualized INFORMATION NEED execution[51]. In the case that this set is empty, we would have a *local* information need, since all global process information would have to be transported into the actual ACTIVITY through the specified CONTEXT VARIABLES.
- „PURPOSE will be reached better" is a shorthand for: In the case that the associated PURPOSE is an OBJECTIVE, i.e., a PURPOSE with a defined measure, the probability shall be increased that a better outcome with respect to the applicable measure will be produced. In the case that the PURPOSE is a non-quantifiable factor, i.e. a GOAL , a MISSION, or a VISION, the probability shall be increased that the PURPOSE can be achieved or approached. This is an informal, but generalized version of the definition of *usefulness* proposed by [Holz, 2002].

---

[51] This property makes sense for a clear definition of our concepts and their relationships. However, in practical applications, it may become irrelevant, because technically, we could give all software components

A **BUSINESS PROCESS SPECIFICATION** (or, a **BUSINESS PROCESS DEFINITION** or **BUSINESS PROCESS MODEL - BPM**) is a SPECIFICATION of a BUSINESS PROCESS with an INTENDED PURPOSE. A BPM is intended to be or is capable of being EXECUTED more than once. Typically, the reusability in various forms at different times is achieved by parameterisation through VARIABLEs. Hence, a BPM can be seen as a SPECIFICATION *schema*.

The SPECIFICATION comprises – in a formal language – specifications of all relevant aspects (in an explicit or an implicit manner) of a BUSINESS PROCESS, i.e. ACTIVITIES, CONDITIONS, RESOURCES, PURPOSE, ACTORS, etc.

To make life a bit easier, we will refer to BUSINESS PROCESS SPECIFICATIONs in the context of this thesis also with the terms PROCESS SPECIFICATION, PROCESS DEFINITION or PROCESS MODEL.

Organization model and resource model

PROCESS MODELs, as well as ACTIVITY SPECIFICATIONs are usually accompanied and complemented by an Organization and / or a Resource Model. These are formal, computer-based representations of the involved ENTITIES and the RELATIONSHIPs between ORGANIZATIONAL UNITS, between OUs, ORs, and OPs, as well as between PERSONs and the three aforementioned kinds of ENTITIES.

An ACTIVITY SPECIFICATION often refers to such a separately provided Organization and / or Resource Model. For example, the ACTOR could be indirectly specified, by reference to an ORGANIZATIONAL ROLE or ORGANIZATIONAL POSITION. [52]

"Explicit or implicit manner" of specifying certain elements refers to the fact that a specification is normally done in some formalized specification language (ideally,

reading access to all data structures, including the totality of potential context information.

[52] Please note that we – though trying to be as consistent as possible with the AIAI's Enterprise Ontology – use here the concept ROLE as a part of the Organizational Model, basically for characterizing a group of PERSONs in an organization with same rights and obligations; not like [Uschold et al., 1998] – as a part of the Meta Ontology – for the way in which an ENTITY participates in a RELATIONSHIP. We consider [Uschold et al., 1998]'s modeling approach in principle convincing and concise, but chose a simpler approach to be more consistent with the widespread terminology in the workflow and business process area. For the future, some terminology alignment and "terminology cleaning" on a clear ontological basis (i.e., a core ontology of business processes and enterprise concepts, consistent with the usual conventions in industry and business science) might be a promising idea.

but not always, with a formal, machine-interpretable, unambigous semantics). Following [WfMC, 1999], a PROCESS DEFINITION shall be represented "in a form which supports automated manipulation, such as modelling, or enactment by a workflow management system." In the literature, at least three major business-process modeling paradigms can be found:[53]

Business-process modeling paradigms

- **Activity-based modeling** is probably the most widespread approach. Here, process models are composed from activity models (specifications), along with product and information flow between activities, as well as some specification of control flow.
- **Communication-based modeling** models business processes as communication acts between performers and customers.
- **Artifact-based modeling** is centered around products (normally, documents) on which operations can be performed as they pass through a series of activities.

In principle, all these modelling paradigms are equally powerful since all of them are able to express arbitrary business processes. However, the different approaches take different perspectives for process analysis and provide different primitives for process modelling. Even within a certain modelling paradigm, there exists normally a multitude of different concrete modelling languages. Often, these modelling paradigms correspond to existing programming language paradigms, such as procedural, object-oriented, or rule-based languages.

Obviously, the decision for one of those process modelling (which also includes: activity specification) languages leads to the fact that not all the properties required by the above definition can be seen directly and explicitly from a formal process specification.For instance, when using the widespread modeling paradigm of Event-Driven Process Chains (EPC, cp. [Scheer, 2001; Aalst, 1999; Nüttgens & Rump, 2002]), each activity is preceded and followed by an event. Further, although it is nowhere modeled explicitly, the semantics of the EPC approach states that the preceding event must have been happened before the activity may be entered. Thus we have an implicit (part of a) PRE-CONDITION.

---

[53] Comprehensive surveys on this topic can be found in [Georgakopoulos et al., 1995; Bach, 1997; Georgakopoulos & Tsalgatidou, 1998; Mentzas, 1999; Myers & Berry, 1999].

Now coming to the KM specific part:

An **EXTENDED PROCESS SPECIFICATION** or an **EXTENDED BPM** is a PROCESS SPECIFICATION which contains at least one KIT SPECIFICATION plus the potentially associated DECISION VARIABLEs and CONTEXT VARIABLEs.

In order to make the overall picture a bit clearer, think back to Figure 23. Those notions are now complemented and extended by the relationships shown in Figure 24.

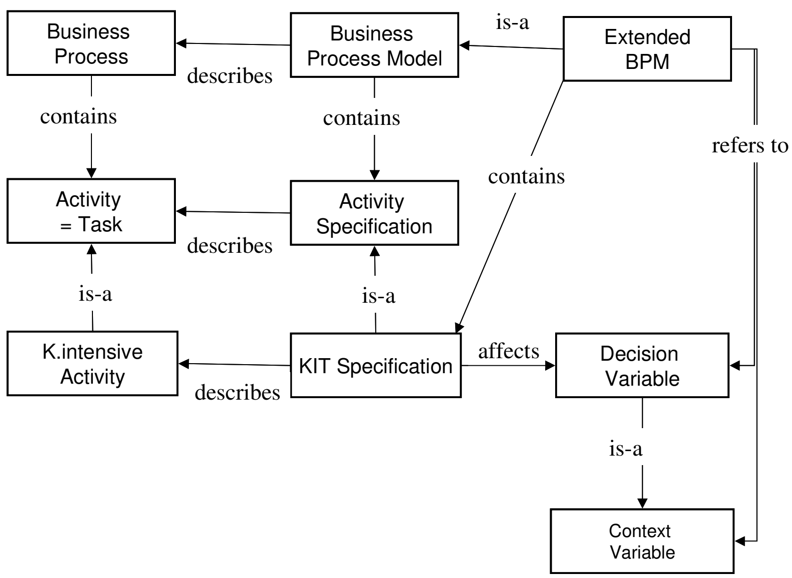


Figure 24: Real World and Model World

At the left hand side of the figure, we see concrete, real activities and processes which may happen or be enacted in the real world. In our model world, such concrete activities are represented by specifications and models which abstract from conrete events and can be instantiated multiple times. Normally, this abstraction effect is achieved by using variables. The figure now also shows the decision and context variables which are introduced exclusively for describing knowledge-intensive activities and their support requests. These variables together with the KIT specifications that contain the support requests, are the extensions which define an Extended BPM as a new class of models, dedicated for expressing KM particularities.

3.1.3.3 Definition of Passive Entities

Here, we only cover the passive entities required for defining OMIS specific contexts and data. We follow [Uschold et al., 1998] with a simple resource definition:

A **RESOURCE** is the RELATIONAL ROLE of an ENTITY in a RELATIONSHIP with an ACTIVITY whereby the ENTITY is or can be used or consumed during the performance of the ACTIVITY.

Now we can adopt the Workflow Management Coalition's definitions for several classes of DATA which can be found in a workflow scenario:

DATA is the universe of all VARIABLES and DATA OBJECTS created, stored, accessed, and manipulated by our OMIS software.

WORKFLOW CONTROL DATA is DATA that is managed by the WfMS and / or a Workflow Engine. Such DATA is internal to the WfMS and is normally not accessible to applications. Nevertheless, some DATA such as instance or activity identifiers may be accessible.

Further, **WORKFLOW RELEVANT DATA** is DATA used by a WfMS to determine the state transitions of a workflow instance, for example within PRE-CONDITIONs, EFFECTs, or for Workflow Participant assignment. Moreover, seen from the perspective of the WfMS, we can distinguish between *typed data* – where the structure of the DATA is implied by its type and the WfMS will understand this structure and will be able to process it – and *untyped data* – where the data structure is not understood by the WfMS and thus it may only be passed to applications.

Last, we have **WORKFLOW APPLICATION DATA** which is application specific and not accessible for the WfMS.

These definitions can be extended by OMIS-specific notions for modeling the dynamic task context of information needs and transporting it to the Knowledge Broker Layer:

A **DECISION VARIABLE** is a VARIABLE, which belongs to the WORKFLOW RELEVANT DATA and is manipulated – normally by a PERSON – in one or more KNOWLEDGE-INTENSIVE ACTIVITIES for achieving the main goals of this activity in the context of the overall process. The manipulation of it might be supported by answering an associated SUPPORT REQUEST.

A **CONTEXT VARIABLE** is a VARIABLE, which belongs to the WORKFLOW RELEVANT DATA and which might be manipulated – normally by an ACTOR – in one or more ACTIVITIES or KNOWLEDGE-INTENSIVE ACTIVITIES. A CONTEXT VARIABLE is needed by at least one SUPPORT REQUEST to express CONDITIONS or INFO NEEDS.

We did not yet define ontologies for specifying background knowledge of an application domain (this will be done in Subsection 3.3). However, given a SEMANTIC VARIABLE would be a variable the codomain of which is determined by a concept modelled in an ontology, then the following statements could be made, insofar: I

We see that a DECISION VARIABLE *may* be a SEMANTIC VARIABLE, but *has not to be* one. In the KnowMore purchasing example, the PRODUCT_TYPE variable was embedded into a „software and hardware products ontology" what allowed to draw inferences and to support retrieval intelligently. However, there will certainly be also CONTEXT and DECISION VARIABLES which cannot or have not to be semantically backed. As a most simple example, the number of products to be purchased might be highly relevant for selecting the appropriate supplier, but there is no need at all for creating an ontology about natural numbers. The same would hold for start data or time of a given process instance. In particular, we have to see the notion of CONTEXT must be relatively broad since there might be many data and information items which might carry interesting context information for a specific functionality in a specific application, but are not ontologically substantiated at all. This comprises also workflow control data or specific business objects manipulated in a given workflow instance. This might even comprise workflow audit data, i.e. log files of recent process executions which are stored in workflow-internal data structures.

3.1.3.4 Definition of Conditionals

A **STATE OF AFFAIRS (SOA)** is a situation. The following is necessarily true for an SOA:

- It consists of a set of RELATIONSHIPS between particular ENTITIES.

- For a particular point in time, the SOA can be said to hold, or be true (or, conversely not to hold or to be false).

ACHIEVE is the realisation of a STATE OF AFFAIRS, i.e. being made true.

An **EVENT** is something that happens or is done at a particular timepoint or over some time interval which has some observable EFFECT in the domain of interest, i.e. it changes the STATE OF AFFAIRS.

Note that we had to change the AIAI's definition of an EVENT (there, it i sdefined as: a kind of ACTIVITY, maybe without a DURATION, an ACTOR, without PRE-CONDITIONs, but with EFFECTs) as a consequence of our more workflow and WfMC-oriented definition of an ACTIVITY (as a logical working step within a PROCESS, done manually or by a machine). We think it makes more sense – also for more generally being understood and accepted – to make the difference between ACTIVITIES as the main active elements to be dealt with in a process execution, and EVENTs which are often also modelled for describing triggers, signals, and notifications coming from outside the automated process modeling and enactment world.

A **CONDITION** is a set of formal statements about the domain of interest which can be evaluated by the software system for a given point of time to be true or false.

- Normally, a CONDITION is a partial characterization of a STATE OF AFFAIRS , typically making equality and comparison statements between VARIABLES and DATA OBJECTS.

3.1.4 Formalizing the Application Layer

Now we are ready to define functions and elements of the OMIS Application Layer (AL) more concretely. In particular, The OMIS Application Layer (AL) has the purpose of providing to human users knowledge-work management functionalities which include:

(1) Process management functionalities:

- Modeling and representation of EXTENDED PROCESS SPECIFICATIONS, their constituent ACTIVITY SPECIFICATIONS, RESOURCE MODELs and ORGANIZATIONAL MODELs.
- Selection and instantiation of PROCESS SPECIFICATIONS and in particular EXTENDED BPMs in response to a user request or key events.
- Interpreting the business logic induced by CONDITIONs in the EXTENDED BPM such that all users logged into the system get offered those tasks of running PROCESSES that they may enact, depending on their ROLE and POSITION.

(2) Activity support functionalities:

- Managing the RESOURCES and the DATA flow such that the AGENTs enacting a TASK are provided with all input and tools they need for this TASK.
- Handing over all SUPPORT REQUESTs to the Knowledge Broker Layer in order to start the execution of INFO NEEDS. Continuous monitoring of global and local task CONTEXT in order to find out changes in CONTEXT VARIABLES (typically, due to user actions) and pass these changes through to the Knowledge Broker Layer.

Before we introduce a generic software architecture to realize these functionalities, let us formally define the interfaces we need for the other OMIS layers.

- Be given:

  a Knowledge Description Layer

$$KDL = (KDL\text{-}Descriptions, KDL\text{-}Background, KDL\text{-}Services).$$

We define a KDL –compliant Application Layer as a quadrupel:

AL = (Context-Services, Interface-Services, KDL, Support-Requests)
for *KDL*

where:

Context-Services is the set of all services offered for querying or manipulating CONTEXT VARIABLES

Interface-Services is the set of all software services available to establish communication with the end user

Support-Requests is the set of all SUPPORT REQUESTs which might be sent during PROCESS execution to the Knowledge Broker Layer

So far, we have defined the Application Layer by the interface services it offers to other OMIS layers, or invokes from other layers, respectively. Some explanations:

Application and Description Layer are closely coupled

- We consider an Application Layer and a Knowledge Description Layer as a relatively fixed unit. Since context and decision variables get their semantics from KDL ontologies, it makes sense not to consider them separately.

What are context services?

- The *Context-Services* comprise functionalities that are typically available to the Knowledge Broker Layer for asking context-related questions when evaluating information needs. These services also include the continuous monitoring of values of context variables and the direct notification of changes in order to react at the Knowledge Broker Layer, if required.

What do interface services?

- The *Interface-Services* characterize the way the Application Layer communicates with the end user. For instance, directly inserting suggested values would be an advanced kind of interface service. At least, the interface services offered by the AL must be able to reasonable present or process the data formats offeredby the Knowledge Broker Layer as results of support requests. As a simple example: it doesn't make sense to retrieve audio files if the AL does not have access to, e.g., a loudspeaker.

Now let us see how one could implement such an Aplication Layer.

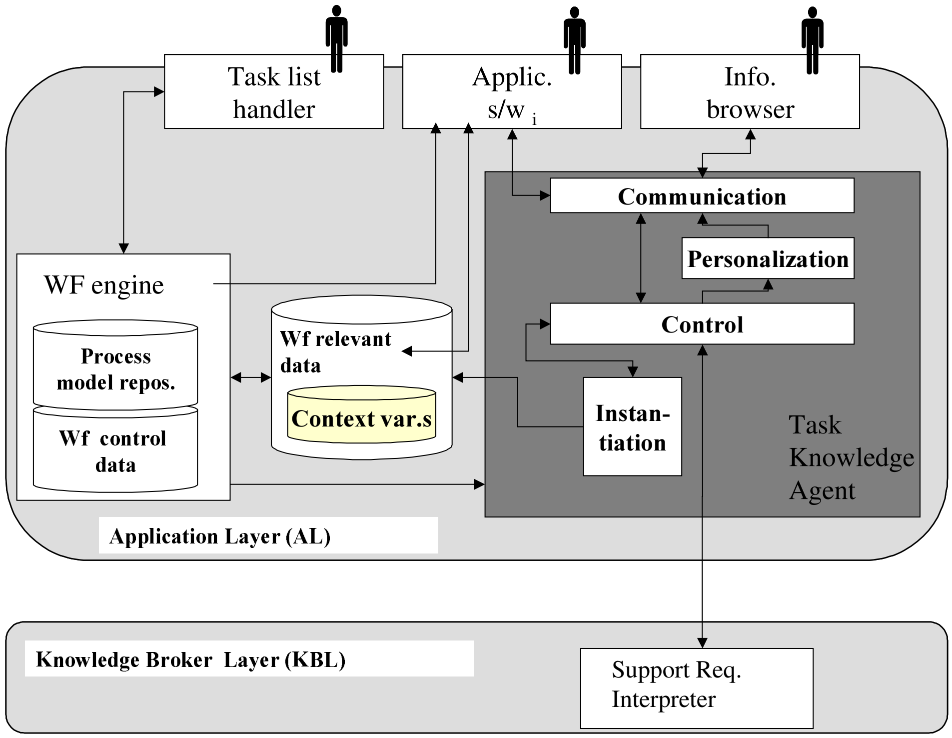


Figure 25: Generic Architecture for Application Layer

Interface elements for the end user:

There are three classes of interface elements accessible by the end user: the task list handler, the information browser, and the interfaces of potentially started applications required for working on some specific task.

The task list handler offers open tasks

The task list handler offers to the user all open tasks in currently active business process instances which could be executed by him/her – taking into account his/her acces rights, organizational role and/or organizational position. Though it is not task handling in the narrower sense, we can also imagine in the same application to have the possibility to start a new process instance, to cancel it or monitor the process status – if we possess the required access rights for this process.

Communication to/from the task list handler

Consequently, the task list handler is an end user interface element. Further, it communicates with the workflow engine (WFE), in the following way:

- Based upon the user's access rights known when the user logs into the system and maintained in a rights management system that we do not show in Figure 25, but can be imagined as a part of the WFE, process names are sent from the WFE to the task list handler in order to allow to start a new process instance.

- During the session of a user, the WFE continuously updates the process status of the process instances owned or controllable by the user. In the other direction, the task list handler may send control information from the end user.
- The WFE continuously updates the list of open task which can be handled by the end user depending on the current process status and the user's roles and positions. Tasks may be deleted from the list when the process instance is canceled or if another user starts to work on the task and it is hence locked for others.
- The logged-in user has the possibility to accept offered tasks from his task list for working on them. This leads to an update of the workflow control data within the WFE and indirectly locks the task for all other users.

Application software is started automatically by the WFE

As it is normal for the middleware function of workflow systems, when a user overtakes a given task, the workflow engine may start task-specific application software (spreadsheets, word processors, specific accounting software, analysis tools, ...) and link it to the appropriate data, documents, and information. Hence, the following communication streams can be identified with respect to these application programs:

Communication to/from the task-specific applications

- Again, the end user may communicate with these application programs in the usul manner.
- Further, there is an initial communication from the WFE that starts and initializes the application programs. Vice versa, a signal must be sent to the WFE when closing the application session in order to proceed with the workflow interpretation.
- The application programs work on the so-called workflow-relevant data-these are, for instance, documents which are routed through the workflow and edited by several people/roles. We consider the decision and context variables of knowledge-intensive tasks to be a subset of the workflow-relevant data since their manipulation is a part of the operational work done in knowledge-intensive activities.
- Finally, there is an important link between application programs and the

communication module of the task-specific knowledge agent. This is exactly where – in the one direction – user behaviour and actions are monitored to find out which decision and context variables are currently interesting to the user and which are their current values. In the other direction, we can think of implanting suggested values directly into the application software's interfaces.

The info browser offers active knowledge services

The third interface element available for the end user is the information browser. As we saw already in the KnowMore example, it will typically contain links to and short descriptions of potentially useful supporting material available in the OMIS knowledge base. However, depending on the kind of support request modeled and on the current system status, we can imagine that there are more kinds of interactions that could make sense, such as:

Possible content presented in the information browser

- Proactively provided support information material or links to colleagues (this is what we said already).
- Browsing or searching facilities in the organizational knowledge base, for exploring networked information or complex topics: It could be the case that an information need states that it makes sense to offer a specific entry point to a carefully selected part of the OMIS content, but leave the exact exploitation of this knowledge resource to the user).
- Input masks for gathering knowledge content from the user: As we will see in the following Chapter 4 about the DECOR solution, it can make sense to formulate a support request that the user should be asked to produce a lesson learned from some decision situation.
- Input facilities for gathering feedback from the user: Since we should aim at a continuous improvement of the OMIS knowledge base (cp. Subsections 3.2 and 3.5), it could make sense that in certain situations the end user is asked – either because this is explicitly modeled in the support request or because the KBL's learning module detects some internal information need – for feedback. This may include:
    - Feedback on retrieval quality (was the knowledge item appropriate for the task at hand?)
    - Feedback on content quality (was the knowledge formulated in the

knowledge item useful, correct, reliable, comprehensive, ...?)

As a side remark it should be noted that feedback gathering might not only comprise getting direct, explicit feedback from the end user. In advanced OMIS implementations we can also think of User Tracking and Analysis modules which monitor user behaviour with the information browser (which links are considered or ignored, which documents are read how long, ...) in order to turn it into system improvement suggestions (see, for instance, [Srivastava et al., 2000; Berendt et al., 2003]).

The following relationships to other architectural elements can be identified:

Communication to/from the information browser

- Of course, the information browser "communicates" with the end user.
- Besides this, the task-specific knowledge agent i the only system module linked to the info browser. It provides the browser with links and other information to be rendered, and it receives feedback information etc.

Now we come to the "internal" elements of the Application Layer, which are the workflow engine, the task-specific knowledge agents, and the respective databases.

The workflow engine controls overall system behaviour

The "conventional" part of the OMIS Application Layer is the workflow engine which has the following functionalities:

Starting a process

- Based on requests from the end user (in our overview, initiated through the task list handler), it starts new processes. This can also happen on events from outside the system – for instance, some business processes are started by an incoming business letter from some business partner which could be analyzed automatically and piped into the intra-organizational middleware. When starting a new process, the process model is instantiated by the respective detailed data, and the appropriate instantiated data structures are created in the workflow-control database and the workflow-relevant data-base. Then the interpretation of the business logic is started.

Interpreting the process logic

Starting the task knowledge agent

- For each task taken for enactment by a human user, the WFE starts the respective application software with the required data or documents, and it initiates a Task Knowledge Agent – provided it is a knowledge-intensive task.

A Task Knowledge Agent takes the SUPPORT REQUEST associated with a given task in the EXTENDED BPM, instantiates it with actual data, and sends a respective service invocation to the Knowledge Broker Layer. Further:

A task knowledge agent manages one support request for a knowledge-intensive task

- Changes in CONTEXT VARIABLES which affect the actual support request, must be sent to the KBL.
- Results from the KBL must be communicated to the user in an appropriate way.
- This might involve personalization aspects if there is personal profile data available.

The Application Layer contains the following data sources:

- The WFE must hold or have access to a Process Model Repository.
- The WFE must hold or have access to the Workflow Control data mainly determining its concrete behaviour.
- The WFE and the Application Software must have access to Workflow Relevant Data. Here, we subsume CONTEXT and DECISION VARIAB-LES that must also be accessed by the Task Knowledge Agent.

This should be sufficient detail to understand the functioning of the AL. Now let us proceed with the next layer of the OMIS architecture.

3.2 The Knowledge Broker Layer

Knowledge is like money: To be of value it must circulate,
and in circulating it can increase in quantity and, hopefully, in value.
Louis L'Amour

The Knowledge Broker Layer (KBL) is the (active) element of our architecture which bridges between (a) knowledge needs and knowledge creation when working on the application tasks, and (b) the (passive) knowledge stock stored and described in KOL and KDL. Hence the KBL has to provide for the implementation basis for all functions and services that should populate an OMIS framework. Since we want to design our generic framework in such a way, that, in the future, it can – in theory and in practice – be extended in all reasonable directions for creating next generation OMIS systems, it makes sense to think a bit fundamentally about functions and services to be provided by an OMIS. To this end, let us make a short excursion in the surrounding theory.

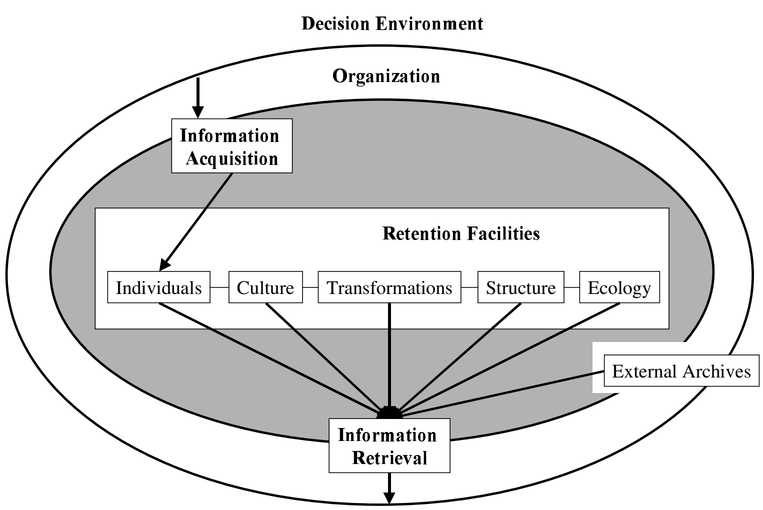


Figure 26: Structure of Organizational Memory, after [Walsh & Ungson, 1991]

3.2.1 Methodological Background

Taking the perspective of Information Science, one can fundamentally analyse what the essential elements of Organizational Memory are and which basic functionalities an OM – and thus also an OMIS – should provide.

Seen from a general point of view, our work can be understood within the general framework of [Walsh & Ungson, 1991] which describes the overall structure of an OM, as shown in Figure 27.

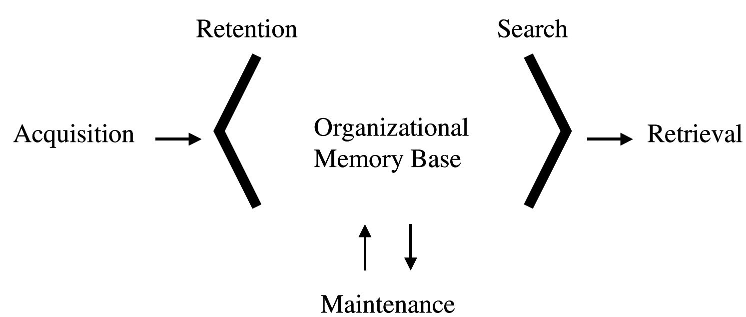


Figure 27: OM Processes, according to [Stein, 1995]

Besides the fact that this picture reminds us that Organizational Memory contains far more than only explicit information stored in computer or other archive systems, like individual knowledge and knowledge stored in group processes or in organizational culture, it also introduces the notion of fundamental *mnemonic processes* offered by an OM. In this first simple model, only acquisition, retention, and retrieval are mentioned. After their seminal work, many researchers further developed OM models; for instance, [Stein, 1995; Stein & Zwass, 1995] added the crucial process of maintenance (see Figure 27).

Mnemonic processes provide the basic functionalities of a memory system

Based on a comprehensive review of the relevant OM literature (cp. [Klamma & Jarke, 1999; Klamma & Schlaphof, 2000],), the list of widespread mnemonic processes shown in Table 15 (translated from [Klamma, 2000]) was derived. Based upon this theoretical analysis and some interesting case studies from the area of continuous quality improvement in several application domains, Klamma came up with three major groups of mnemonic processes, and a number of conrete, practice oriented examples for each of these groups.

Reference	Mnemonic Processes
[Stein & Zwass, 1995]	Acquisition, Search, Retrieval, Retention, Maintenance
[Probst et al., 1997]	Knowledge goal definition, Identification, Acquisition, Development, Sharing, Retention, Usage, Evaluation
[Ramesh, 1997]	Acquisition, Retrieval, Retention, Development, Reuse
[Morrison, 1997]	Acquisition, Search, Retrieval, Retention
[Wijnhoven, 1998]	Acquisition, Search, Retrieval, Retention, Dissemination
[Burstein et al., 1998]	Acquisition, Retrieval, Retention, Maintenance, Learning

Table 15: Mnemonic OM Processes, adapted from [Klamma, 2000]

Since we consider both the idea of a fundamental organization and clustering reasonable, and the concrete examples useful, we will quote these results almost literally. The major process groups, according to Klamma & colleagues, were:[54]

Three basic categories of mnemonic processes in an OMIS

- **Knowledge creation and acquisition:** containing the specific processes archival acquisition; directed acquisition; automatic acquisition.
- **Knowledge search, retrieval, and usage:** containing the specific processes asking an expert; query; guided exploration; filtering; navigation; subscription; task-specific publishing.
- **Maintenance:** containing the specific processes integration (which shall mean administrative functions – such as rebuilding an index – which must be performed for inserting new knowledge into the memory); aging (which shall mean to monitor actuality of knowledge and explicitly dealing with this topic – e.g., think about explicit forgetting); reorganization; validation (recognize meaningfulness of knowledge objects and handle not meaningful knowledge objects).

---

[54] Since Klamma's thesis [Klamma, 2000] is in German, and some of the terminology is not easy to translate exactly (which is even more subtle, since the German terms are obviously often translations of the English original literature), and – moreover – the same process groups have slightly different names in [Klamma & Jarke, 1999], we refer the reader who is intersted in subtle details, to the original literature by Klamma & colleagues, as well as their sources. For this presentation here, we preferred just to create compounds group names which combine aspects of the [Klamma, 2000] and the [Klamma & Jarke, 1999] terminology.

Klamma provides reference processes for selected mnemonic functions.

As already mentioned, the procedure seems insightful, and the concrete examples, as well as their implementation by [Klamma, 2000] in the form of *reference processes*, is useful. Nevertheless, when having a closer look at the – only informally defined – process groups, one can encounter easily problems to come up with a clear definition of these process groups, which would be useful to have, e.g., for an unambiguous classification of *new* mnemonic processes. Hence it makes sense to add some clarifications. Moreover, it might make sense to add some functionalities not yet mentioned, and to apply some restructuring. In detail, we extend and explain Klamma's approach as follows:

Acquisition and integration seem a bit confusing

- At first sight, it looks strange to mention acquisition ("place new knowledge in the knowledge base") and integration ("organize a meaningful introduction of new knowledge objects") as two separate processes, even in different process groups. However, this distinction can be explained by a separate treatment of processes changing the Knowledge Object Level (e.g., one – for instance some "normal knowledge worker" adds a document to the Intranet) and processes changing the Knowledge Description Level (e.g., later – a knowledge manager or subject area editor – creates the metadata for this document). This might make sense in some scenarios where we could imagine the two steps performed by different organizational roles at different points of time. So, we should notice:
    - A generic architecture and definitional framework should adopt this two-step approach.
    - Concrete, real-world processes may often combine both activities in virtually one step.

Now, let acquisition extend the KOL content, and maintenance extend or change the KDL metadata

- This might lead to the clarification that all maintenance processes are exclusively dealing with the metadata. This makes sense, since in Computer Science it is a comprehensible argument to refer the term "maintenance" more or less to "system knowledge", typically code, or system-internal data. Some remarks to this decision:
    - We are not sure that this is in the spirit of Klamma et al., who, for instance, mentions the dealing with outdated or invalidated knowledge objects also as part of the maintenance processes. But, for such a task it might be absolutely reasonable to delete such knowledge objects

from the knowledge base, hence changing both the knowledge base and the metadata.

- Another explanation for the distinction between acquisition and maintenance could be that the two activities might be performed by different roles in the KM organization structure (cp. [Klamma & Jarke, 1999]): acquisition by the Knowledge Creator role – typically knowledge workers in some subject area, integration only by the Knowledge Administrator role, typically some person resonsible for the knowledge stock in some topic area. But then, it is not that convincing to have the aging and validation process only available to the Knowledge Administrator, since the Knowledge Creators could in some scenarios equally well, or better, recognize outdated knowledge objects. Why should the creator of a knowledge object not be allowed to delete it also? But, if this was allowed, he would need access to the Adminstrator role, or we would have to introduce, a "delete" process for the knowledge objects.

A KM organization has also to think about KM roles

- After all, a completely consistent interpretation of Klamma's suggestion was difficult for us, when enforcing a strict formal definition of process groups. So, we stay with our initial idea to refer the acquisition processes to KOL and Knowledge Creators, and the integration process to the KDL and a Knowledge Administrator; this works if we add the assumption that no knowledge object will ever be deleted actively from the KOL. Of course, deletion of knowledge objects is not necessary, if the metadata contain a suited attribute value "outdated" or "invalid". Even if this might sound unrealistic in practice, it is theoretically sound, not far away from real, biological memories, and does not destroy any possibilities, e.g., for traceing old decisions, for analysing past trends and developments, etc.

Referring acquisition and integration to KOL and KDL, resp., seems methodologically intelligible – and can be kept up if aging and invalidation affects only KDL metadata .

- Now that we raised the point that there are processes *acquiring content* and processes *changing the metadata* (at the system level: reorganisation, at the single object level: aging, validation), it becomes obvious that there should probably also be processes *changing the content*. For instance, if some Knowledge Creator wants to refine or correct an entry which she inserted in the

Why not allow content change, if we allow metadata change?

organizational knowledge base. Technically, this can also be described within the frame already specified by Klamma *et al.*, namely if changing means inserting a new object and linking it to the original version using a special metadata entry for storing the history of knowledge objects. This is also consistent with our view of aging above: nothing will be deleted or persistently changed, instead we add new objects and create a change log. So far, our discussion of Klamma's concepts did not bring a fundamental critique, but may lead to two technical basic decisions:

We should allow for basic and compound mnemonic oprators

- We don't have to introduce many new operators. But it could make sense to introduce *compound mnemonic processes* (that might affect both knowledge objects and knowledge descriptions) which consist of several steps which are basic, *atomar mnemonic processes* (that affect either knowledge objects or knowledge descriptions / metadata).

Let us call them: mnemonic micro processes and mnemonic macro processes

  - Let us also call the latter *mnemonic micro processes* and the former *mnemonic macro processes*.

- If we think about a minimal theoretical basis – which also leads directly to implementation decisions – for such basic building blocks of mnemonic processes, we conceive:

  1. There are only two elementary kinds of entities which have to be stored as DATA OBJECTS in our system: knowledge objects (content/documents in the KOL) and knowledge descriptions (metadata objects in the KDL)

  Elementary entities stored in the OMIS

  Elementary operations on such entities

  2. There are only three menomonic micro processes that are necessary, i.e. elementary operations which manipulate the elementary DATA OBJECTS:

Create a knowledge object

     - Create_knowledge_object:

       Input: a document, optional: an Information Source

       Effect: the document ist stored in (the specified) Information Source (or a source determined by the OMIS)

       Output: a knowledge object

- Create_metadata_object:

  Create a metadata object

  Input: a metadata object type, a (set of) property value pairs for properties of this metadata object type where the property values may be basic data type elements, or data object id's identifying knowledge objects or metadata objects

  Effect: the respective metadata object is stored in the KDL

  Output: a metadata object

- Query_KDL:

  Access the knowledge description layer

  Input: a metadata object type, a (set of) search constraints (restrictions on property values for certain properties of this metadata object type), optional: a property name for this metadata object type $name_i$

  Effect: ./.

  Output: the (set of) metadata objects which fulfill the search constraints, or, respectively, their property values for property $name_i$

3. All other functionalities of the OMIS memory part can be realized as mnemonic macro operators[55] assuming that there exists some sufficiently expressive macro language, e.g., equivalent to some simple procedural programming language or flowchart approach which contains at least:

   Everything else can be realized as mnemonic macros

   - chaining of (micro, or other macro) operators, conditional expressions, some looping mechanism

4. Now, coming back to Klamma's terminology, we can say:

   - all mnemonic operators that contain only knowledge object creation and maybe KDL querying can be called knowledge creation or acquisition process

---

[55] As an example take the attachment of a note or memo to a document. If this is part of personal knowledge management, it would consist of two elementar processes: (1) the memo document is stored in the content knowledge base of the KOL which is *knowledge acquisition*; (2) the suited metadata are created and stored in the KBL which describe the link between original document and memo document, as well as bibliographic and other metadata which can be derived from the creation situation. This is *knowledge integration*, i.e. a part of maintenance. If such a memo is not only part of the personal management, but embedded into some group KM process – e.g., in a research department where discussions about new technology trends may be part of the daily work – a further step (3) could be imagined: inform other, potentially interested colleagues to join the discussion. This could be understood as *knowledge usage*, potentially leading to a new *knowledge acquisition* cycle.

- all mnemonic operators that contain only metadata object creation and maybe KDL querying can be called maintenance process
- all mnemonic operators that contain only KDL querying can be called knowledge usage process
- moreover, let us consider all mnemonic operators being knowledge maintenance processes which manipulate – for instance by learning from user feedback – only KDL- or AL-internal knowledge (such as task-specific information needs, user profiles or an info agent's rerieval knowledge), but not the KOL.

KM processes are business processes aiming at KM goals, containing KM Tasks, which realize mnemonic functions

5. Further developing these considerations, we can compose even more complex mnemonic processes as workflows at the Application Level. Then we would – outside the Knowledge Broker Layer – call a process a **KM process** or **Knowledge Process** if (i) it pursues primarily goals concerning the organizational KM strategy and aims at knowledge creation, knowledge maintenance, or facilitating knowledge usage objectives; (ii) it is primarily or exclusively enacted by KM roles in the organization; and (iii) if its tasks are to a significant extent KM tasks.

6. Typical KM processes are editorial processes for Intranet content, best practices, etc.[56]

Some functionalities might be realized either as KM process or as a mnemonic macro.

7. Then, of course, there is some degree of freedom / ambiguity in our model: it is not unambiguous whether a certain functionality should be realized as a KM process or as a mnemonic macro. This seems acceptable as a space for design decisions. There are some indicators: functionalities which are enacted by (several) humans, over a longer period of time, maybe involving many different kinds of mnemonic

[56] And it is an interesting – and useful – research topic to think about reference KM processes. Considerations in this directions – at least, interesting, transferable sample implementations – can be found, for instance in work done at the Hochschule St. Gallen's work (e.g. [Schmid et al., 1999]). A further topic would be to integrate such processes with industry or official standards for Quality (cp. ISO 9000, EKMF) or Service Management (such as ITIL , the IT Infrastructure Library, see http://www.itil-itsm-world.com/ [Last access: 04/15/2004]), or combine this with KM Maturity Models (cp. [Hefke & Trunko, 2002 ; Paulzen & Perc, 2000 ; Oberweis & Paulzen, 2003]).

operators, might be rather a KM process.

- Now coming back to our initial goal of discussing Klamma's approach, we can further find that the topic of machine learning for improving OMIS content is covered only cursory, or implicit, respectively. He is mainly focussing on process knowledge. Our framework should elaborate in a bit more fundamental manner all relevant learning processes in an OMIS,

An OMIS framework should explicitly point out automated learning and self-organization opportunities

Since it became obvious that mnemonic processes should become the elementar functional building blocks of an OMIS KBL, let us – in the light of our considerations made so far – summarize Klamma's list of mnemonic functions, with some extensions coming from our own experience or other literature, and some adaptations / clarifications coming from our discussions made above. This summary is shown in Table 16, Table 17, and Table 18 [57], respectively.

Mnemonic process dealing primarily with knowledge acquisition	Micro or macro process	Mnemonic process group
Archival acquisition: knowledge agents places KO [Morrison, 1997] – comprises also notes attached to KOs, discussion threads around KOs or KIDs	micro	KA
Contextualized storage: automatic knowledge agent stores KOs with known process context [Abecker et al., 1999b]	macro	KA + KM
Directed acquisition: a knowledge agent is asked to insert some knowledge [Morrison, 1997] – e.g., an unsucessful search triggers the creation of the missing KO	micro	KA
Automatic acquisition: periodically, knowledge agents are triggered to extend the KOL [Morrison, 1997] – contains, e.g., Data Mining functionalities	micro	KA

[57] Please note that some of those functionalities abstract from some details, thus representing whole classes of concrete, implementable mnemonic functions. We should also note that some terms are not clearly defined in relation to each other. For instance, filtering, subscription and task-specific push are difficult to distinguish. Nevertheless, we kept these terms for illustration purposes since they have some intuitive semantics and may help to draw the picture of how a fully populated Knowledge Broker Layer could look like. ...

Critiquing: if the OMIS itself possesses some problem-solving expertise, it might detect "interesting" actions of the user and ask "why did you do that?" question to extend its own knowledge [Fischer et al., 1991 ; Qiu & Riesbeck, 2004]	macro	KU / KA

Table 16: Knowledge Acquisition Mnemonic Processes

Mnemonic process dealing primarily with knowledge usage and retrieval	Micro or macro process	Mne-monic process group
Asking an expert: queries are routed to persons who are experts in the field [Morrison, 1997]	micro	KU
Query: according to the query constraints, externalize specific parts of the OMIS knowledge base [Morrison, 1997]	micro	KU
Guided exploration: search is supported by a search expert, e.g. a software agent [Morrison, 1997]	macro	KU
Filtering: restrictive filtering of query results in order to present the user-relevant KOs only [Morrison, 1997]	macro	KU
Navigation: metadata are filtered, combined and presented in such a way that the user can navigate through appropriately visualized knowledge structures	macro	KU
Combined querying and browsing: mixed-interaction interface for information search by knowledge agents	macro	KU
Personal subscription: according to a predefined query, KOs are sent periodically to a knowledge agent [Morrison, 1997]	micro	KU
Task-specific push: route relevant KOs to the task-enacting knowledge agents [Ackermann & Mandel, 1995; Klamma et al., 1998]	macro	KU
Role-specific push: route relevant KOs to the knowledge agents holding specific organizational roles [Schmalhofer & Elst, 1999; Elst & Abecker, 2001]	macro	KU
Position-specific push: route relevant KOs to the knowledge agents holding specific organizational positions	macro	KU
Reminding: depending on time, contextual factors or other, specified conditions, bring certain KOs proactively to the user's mind	micro	KU
Critiquing: if the OMIS itself possesses some problem-solving expertise, it might detect "interesting" actions of the user and ask "why did you do that?" question to extend its own knowledge [Fischer, 1991; Qiu & Riesbeck, 2004]	macro	KU / KA / KM
Contextualization: techniques such as (i) presentation enrichment, (ii) aggregation, linking, and unification of KOs, as well as (iii) visualization of links and relationships between KIDs, KOs, and context elements are used for (a) improving comprehension, (b) reducing information overload, (c) guiding association of KOs, or (d) supporting comparability [Klemke, 2002]	macro	KU

Task-, role, or user specific summarization: it might make sense to summarize long texts before presentations; here, "shrinking factor" and topical focus can depend on the person, role, or task	macro	KU

Table 17: Mnemonic Processes Dealing Primarily with Knowledge Usage

Mnemonic process dealing primarily with knowledge maintenance	Micro or macro process	Mne-monic process group
Integration: integrate new KO in the KDL and (maybe) perform depending actions [Klamma, 2002]	macro	KM
Aging: monitor usefulness of KOs and treat aged knowledge [Klamma, 2002][58]	macro	KM
Reorganization: KOs are re-organized, i.e. KIDs are changed, according to new insights in content domain or in search behaviour, etc. [Klamma, 2002]	macro	KM
Validation: recognize meaningful KOs and treat unmeaningful ones [Klamma, 2002][59]	macro	KM
Adapt user model: use direct and/or indirect search feedback and / or other data (e.g. other user's behaviour, in Recommender-like approaches) for improving a personal user profile	macro	KM
Metadata adaptation: when ontologies change in the KDL, it might make necessary to change depending metadata in turn [Stojanovic et al., 2002]	macro	KM
Usage-driven ontology evolution: observing user search logs, as well as query feedback, it is possible to detect deficiencies in the current ontologies as a knowledge organization tool [Stojanovic & Stojanovic, 2002]	macro	KM
Corpus-driven ontology evolution: depending on the texts of KOs in the KOL, ontology evolution may be triggered (new terms occur, meanings shift, ...)	macro	KM
Adapt retrieval knowledge: user behaviour, explicit user feedback, and communication among different retrieval agents may lead to learning of new or better retrieval knowledge (cp., e.g. [Wess, 1996; Rocchio, 1971; Hust, 2004]).	micro	KM

Table 18: Mnemonic Processes Dealing Primarily With Knowledge Maintenance

Abbreviations used in the tables:

- KO: knowledge object (e.g., a technical report, a memo, an audio record, a technical drawing, an email, a powerpoint presentation, a CBT simulation software, ...)

[58] Here we see a slight terminological sloppiness : though [Klamma, 2002] talk about "usefulness" in general, he only refers to age in the name of the process. This should be understood much more general, of course.

[59] This seems again a somehow vague definition. Probably, one should combine aging and validation and talk about quality monitoring.

- KOL: Knowledge Object Layer, the archives in the narrower sense (without KM-specific metadata)
- KA / KU / KM: Knowledge acquisition, Knowledge Usage, Knowledge Maintenance process group
- KID: Knowledge Item Description, i.e. a metadata object stored in the KDL

In order to close our general discussion about mnemonic operators and functionalities to be offered by the Knowledge Broker Layer, we summarize from a bit different perspective, as shown in Figure 28. Similar to Klamma's three groups of processes, we visualize basic functionalities in a way which comes a bit closer to usual presentations (cp. Figure 3):

Acquisition processes at the top

- At the top, we have mnemonic processes and system functionalities mainly interacting directly with the user, i.e. the knowledge acquisition pocesses and manifold kinds of collaborations between knowledge agents (e.g. when "asking an expert" establishes a contact between two people) or between people and the system (e.g. when a "combined query / browsing" process guides the user through complex knowledge spaces, or when a "critiquing" components asks an expert for an explanation). Typically, such collaborative aspects are interface issues of other mnemonic processes, or they lead to some knowledge acquisition.

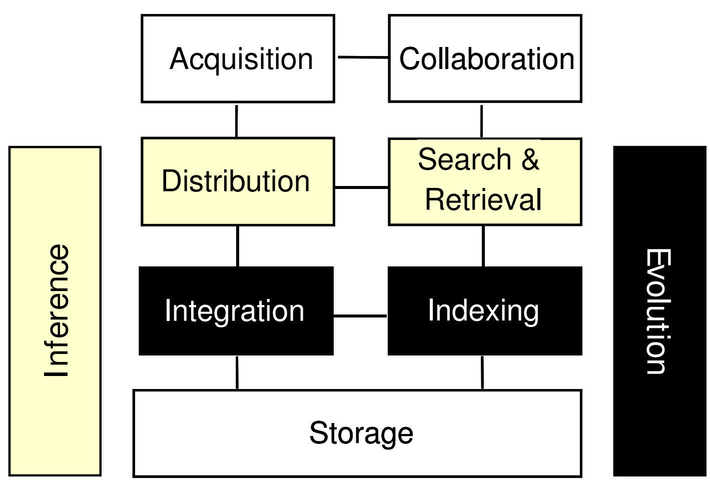


Figure 28: Simplified Overview of Functionalities Hosted Within the Knowledge Broker Layer

Knowledge usage in the middle

- The boxes with the light yellow shade (distribution, search & retrieval, inference) are mainly concerned with *knowledge usage* processes. They identify active (push), passive (pull), as well as background processes (inference is normally used within knowledge retrieval algorithms for finding or processing knowledge.

- The black boxes (indexing, integration, evolution) represent knowledge maintenance. For better understanding we kept indexing and integration separately. While the former shall stand for finding the best content desriptor for the metadata object of a KO, the latter means merely establishing links and relationships to other metadata objects. Of course, both could be realized within (or, at lest accessed via) one and the same software module.

Knowledge maintenance in black

So far, our discussion of Klamma's work gave us already some deeper insights about a generic architecture for the KBL realization, as well as some ideas for further menmonic functions not yet covered in his model. Now we will embed these insights into our conceptual framework. But before, let us make the above promised clarification about the relationships between business processes, menmonic processes, etc.

Based upon Klamma's model, we will build our generic architecture

Excursus: Business Processes & Mnemonic Processes.

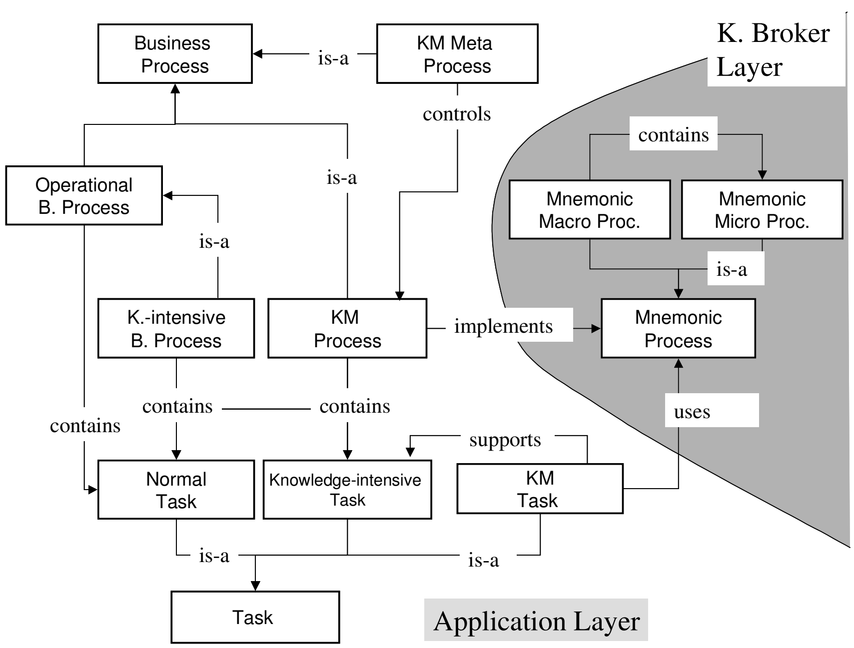


Figure 29: Metamodel Business & Mnemonic Processes

Let us exemplarily explain the relationships between terms and concepts in Figure

29:

> Buying simple office material might be an “ordinary”, operational business process. It contains a number of “normal” tasks, such as filling in forms, paying, sending a letter to the supplier, etc.
>
> Now, buying a first-class personal computer might contain a knowledge-intensive task, namely deciding which product to buy.
>
> With this knowledge-intensive task, the overall purchasing process is then also a knowledge-intensive business process, containing normal and knowledge.intensive tasks.
>
> The knowledge-intensive decision could now be supported by an automatic retrieval and partial analysis of product test reports from the Internet. The corresponding retrieval could be a mnemonic process within the OMIS Knowledge Broker Layer.
>
> In order to link this KBL-internal Mnemonic Process into the purchasing process, we would have to insert a KM task (= KM service, knowledge task) which establishes the link between the operational decision task and the Mnemonic Process, handing over the appropriate context variables, combining it with other Mnemonic Processes, etc.
>
> Here, the explanation could stop. However, in order to go through the whole picture: imagine we want to gather experiences from expensive purchases in a Lessons Learned database. Then we would have to link another Mnemonic Process into the operational process (maybe through the same KM task, maybe through another one), for getting feedback from the user and storing it. If this feedback should be edited and evaluated from time to time (an expert should have a look at the lessons learned database each eight weeks to sort out redundant or outdated entries), we would have to embed this storage task into a whole editorial process. This would now constitute a KM process.
>
> If this KM process would now be monitored each 6 months, because the corporate KM strategy foresees some regular check of Lessons Learned status and rethinking of KM activities, this could be part of a more strategically oriented KM Meta Process.

Now, the stage is prepared for fully formalizing the KBL.

3.2.2 Formalizing the Knowledge Broker Layer

After the extensive discussion of Mnemonic processes, the question arises how to make available such services to the end user located at the Application Layer, and how to link them with information and knowledge in the KDL.

- Be given:

  a Knowledge Description Layer[60]

  KDL = (KDL-Descriptions, KDL-Background, KDL-Services) and

  an associated Application Layer

  AL = (Context-Services, Interface-Services, KDL, Support-Requests)

  for *KDL*

  such that the union of all MNEMONIC PROCESSES
  occuring INFORMATION NEEDS that are
  parts of elements of the set *Support-Requests* is called *MP-Requests*

  and the union of all CONDITIONS used in
  INFORMATION NEEDS that are
  parts of elements of the set *Support-Requests* is called *PPL_Inst*

- Then we can define

  a *Knowledge Broker Layer KBL for KDL + AL* as a quintuple:

  KBL = (MP-Services, PPL,
  Context-Requests, KDL-Requests, Interface-Requests)

  such that:

  MP-Services is an interface which offers a set of services that realize Mnemonic Processes

  MP-Services $\supseteq$ *MP-Requests*

  PPL' is a formal language for formulating conditional expressions
  such that all elements of *PPL_Inst* can be formulated with PPL

  Context-Requests is a set of possible service invocations such that

  Context-Requests $\subseteq$ *Context-Services*

Some explanations:

- As the definition says, the purpose of the KBL is to accept requests for execution of Menominc Processes (what we called Support Requests or Support

[60] Please refer to the definition of the Knowledge Description Layer KDL in Section 3.3.

Specification before) from the Application Layer and implementing them – under the use of context knowledge – as a set of invocations of KDL services.

- The implementation of Mnemonic Processes is done through the orchestration of (a set of) information agent(s) which implement micro and / or macro processes.
- Here we have to note (as we will see in the next Section) that the notion of KDL services is pretty comprehensive, encompassing reading, writing and change of metadata and knowledge objects, as well as reading and changing the whole background knowledge base (ontologies, process models, etc.).
- *MP-Services* is the set of all Mnemonic Processes implemented in the KBL as Info Agents and offered through the KBL service interface.
- The set *MP-Services* must not to be empty since this would mean that the KBL offers no functionality. At least, it must offer some access to a *query* functionality. From a theoretical point of view, it has necessarily to have some *storage* functionality for KOs since we could imagine a "read-only" OMIS where the content is either defined once and then stable, or is only filled from outside the OMIS. E.g., a pure Internet information system would fall into this class.
- *Context-Requests* is the totality of all kinds of services invoked by any KBL, Info Agent, or by the KBL Support Request Interpreter when interpreting support requests. This may comprise investigate process models, ask for workflow-relevant data, ask for user details, etc.
- *KDL-Requests* is the totality of all requests for services of the Knowledge Description Layer invokable when the KDL executes Support Specifications. This means, it comprises all reading or writing operations to the KDL which are required for Mnemonic Micro Processes or Menmonic Macro Processes, including reading / writing / creating metadata objects in the KDL, as well as invoking services which ask for information, changes, or inferences from the Ontology Management System within the KDL.
- The set *KDL-Requests* should not be empty since this would mean that the OMIS does not delivery content, but only metadata – which might be possible

in seldom, very specific cases, but is not the intention of an OMIS. So, normally we would expect at least a KDL query operator to be contained.

- *Interface-Requests* is the totality of all requests for services to be executed by the user interface management of the Application Layer for presenting knowledge and controlling interaction with the end user.
- Further all pre-conditions used in Information Needs and in the Post-Processing specifications of Support Requests must be expressed in a language such that the evaluation of truth values of conditions can be reduced to KBL-internal computations plus KDL-accesses covered by the set of *KDL-Requests* operations.

Please note: while it makes sense that Application Layer and a Knowledge Description Layer are always considered together, one could easily imagine that – standardized interfaces, or, very easy service requests provided – a KBL is replaced by another one which works also well with a given AL-KDL combination.

Now we can derive a generic implementation of the KBL as shown in Figure 30.

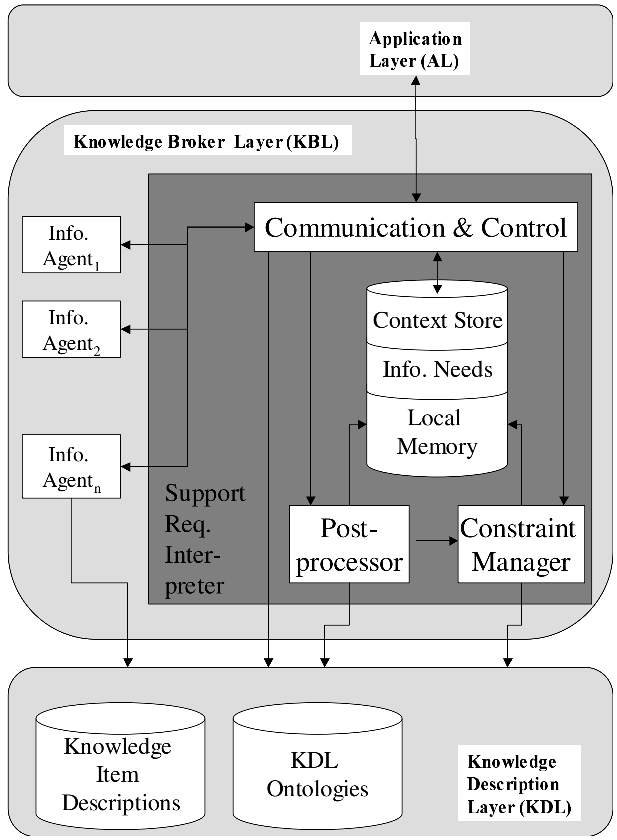


Figure 30: Generic Architecture for Support Request Interpreter

Before we discuss the architectural elements in detail, let us give a short overview:

- The **Support Request Interpreter** (SRI) has the overall control over the KBL for one actual task instantiation.
- The **Communication & Control Module** within the SRI coordinates communication with Application Layer, Knowledge Description Layer, and Info Agents.
- The **Constraint Manager** within the SRI evaluates conditional expressions in the language *PPL* to TRUE or FALSE.
- The **Postprocessor** within the SRI lets evaluate pre-conditions of postprocessing and executes post-processing of KOs and KIDs before sending them for further manipulation to the AL using messages of type *Interface-Requests* and to the KDL, respectively, through messages of type *KDL-Requests*.
- A **Memory Module** internal to the SRI facilitates the other modules.

- The **Info Agents** are a set of software modules realizing Mnemonic Micro and Macro Processes on behalf of the SRI, and communicating with the KDL .

Now let us go into more detail for the several KBL software modules:

The central element of the generic software architecture for the Knowledge Broker Layer KBL is the Support Request Interpreter (SRI). This module is responsible for receiving support requests from the AL, breaking them down into Information Agent calls, combine and postprocess the results and send them back to the Task Knowledge Agent in the Application Layer. In principle, there must be one running SRI instance, or one autonomous SRI thread, respectively, per currently active knowledge-intensive task. The functionalities and sub-modules of the SRI in detail:

The Support Request Interpreter stands at the heart of the KBL

First we have to mention the SRI's internal memory. This holds three kinds of information:

Functionalities and sub-modules of the SRI

- The Context Store keeps all available local and global context information. It is continuously updated by the Application Layer if specific parameters change.[61] Context information is used for:
  - Evaluating pre-conditions in information needs and post-processing.
  - Intelligent retrieval services done by Information Agents – which might use context information (together with the ontologies were the context-variable parameter values are taken from) for query disambiguation, for query rewriting, or for similarity assessments.
  - Creating storage context for Information Agents that realize knowledge archiving services.

Kinds of usage for stored context

[61] From the technical point of view, if we have several SRI instances or SRI threads, this part of the storage area might be shared by all of them. In more distributed scenarios, maybe with other software components that exploit some notion of context, we could even imagine a Context Agent as a relatively independent module which ansers questions via a Context API (Application Programming Interface). Further it should be noted that the Context Store – which is conceptually a coherent module, located at the KBL – might technically also be only a homogeneous interface to parts physically realized in different system areas. In particular, data about global process context (prior and future tasks), about process-instance data (process instance initiator, process instantiation time, etc.), or about the organizational model (relationships between people according to organigram etc.) might technically be realized by sending queries to the Workflow Engine or other elements of the Application Layer.

- The Information Need Store is the storage area where the SRI actually keeps the information sent by the Application Layer when initiating a task-specific support and handing over the detailed support specification:
  - What Information Needs are relevant to this task, how are their parameters actually bound to concrete variable values[62], how are the several Information Needs defined and how do they relate to each other, ...
  - This information is used to delegate tasks to concrete implementations of Information Agents.
  - The third part is an SRI Local Memory for every kind of intermediate results or local variables. In particular, this comprises:

Local data within the Support Request Interpreter

    - results which go into the postprocessing (ordering)
    - intermediate results of some more complex postprocessing (which, e.g. does analyses over content and amount of information retrieved)
    - results sent by some information retrieval Info Agents$_i$, which have to be further processed, be it in the post-processing (ordering), or by some other Info Agent$_k$ which uses the result for enacting its own knowledge service (e.g. because the result of Info Agents$_i$ specifies where Info Agent$_k$ has to search for information, or about what topic to search further)

The Constraint Manager evaluates preconditions for information needs and postprocessing

The next SRI module is the Constraint Manager which evaluates preconditions as specified for Information Needs and for the post-processing. To this end, it accesses the local memory and, if necessary, the background knowledge specified in the ontologies of the KDL.

The concrete precondition languages depends on several environment factors

The design of a concrete precondition language is of course depending from (a) what is really useful in the concrete application scenario; (b) what concrete data can be accessed from the Workflow Engine; (c) which interface does the Ontology Management System in the KDL provide and how do the underlying domain and enterprise ontologies look like; further, what attributes does the Information

[62] Which have also to be updated when changes of context parameters are propagated by the AL.

Ontology provide? (d) which kinds of Information Agents are implemented and what kinds of results do they deliver[63] (e) the question which kinds of user interaction are foreseen even in a mainly pro-active scenario[64] and finally – for the syntax – (f) what is the host programming language and software paradigm where the OMIS is implemented and embedded in? We informally gave some examples in the description of the KnowMore prototype system in Chapter 2. Principally, reasonable preconditions could be imagined which are stated in terms of:

Useful notions for formulating preconditions

- the question whether some context or decision variables have already a value, or not;
- if there is a set of alternatives for a specific decision variable, how many options are contained therein;
- conditional expressions regarding overall process-instance parameters such as starting time or process owner, as well as considerations which involve the organization model that describes the roles and positions of process owner, task actor, etc.;
- the question how many results a specific information need did produce (in particular, whether the results set is empty or has exactly element);
- comparisons between result sets (size, concrete elements) of different information needs with respect to their size or concrete content;
- comparisons between results sets (size, concrete elements) produced by different Info Agents for different Info Needs;
- Comparisons of / selections on / conditions over variable values, talking about KBL-internal variables or context and decision variables;
- arbitrary boolean combinations of conditions like the ones specified above.

---

[63] For instance, it could make a difference for the precondition language for postprocessing whether we get back only an unsorted list of results, or whether a fuzzy retrieval algorithm also produces some relevancy estimation.

[64] For instance, if the user explicitly gives feedback on quality or content of results or if she explicitly asks for more and deeper results, or for specific explanations, this will influence the evaluation, cooperation, and post-processing of information needs.

The Postprocessor is called from the Communication and Control module. It evaluates its preconditions by calling the Constraint Manager and performs actions as specified in the Post-Processing Language for manipulating a result set stored in the SRI's Local Memory.

Possible kinds of post-processing actions

- Grouping (or nested grouping) according to the KIDs attribute values for specific metadata attribute(s).
- Ordering of results according to the attribute values of some metadata attribute which are numeric or possess at least some (partial) order.
- Ordering of results according to some relevance or information quality value returned by an Information Agent.
- Ordering of results according to some accumulated measure computed from values as described in the last two points.
- Post-processing single knowledge objects by calling specialized Information Agents, e.g. for summarizing long texts, for translating foreign languages, etc.
- Post-processing groups of knowledge objects by calling special Information Agents which possess problem-solving knowledge for doing computations, analyses, etc. on results (functionalities that are typically contained in Wiederhold's *mediator agents* [Wiederhold, 1992]).
- Evaluating follow-up information needs, e.g., to find out something about the reliability of the source of a formerly found KO. Or find more details about a topic recognized as relevant. Or, if we have a cooperative information gathering scenario, follow – for further retrieval tasks – the way that some other agent has found[65]

Information agents perform mnemonic processes

Information Agents are generic implementations of mnemonic micro or macro processes. They are invoked with a set of parameters and then – under the control of the SRI and using the SRI as an interface intermediary to the end user – perform knowledge acquisition, knowledge usage, or knowledge maintenance operations.

---

[65] A simple real-world example for this: use the department telephone list to find out the exact name of a colleague; use this name and the telephone book to find out the address of this colleague; use this address and the city map to find the way to this colleague; use this way and your traffic radio to estimate the time to go there; etc.

This means in particular:

- For information delivery, they realize retrieval, filtering, push services etc. Further kinds of knowledge usage may comprise complex problem-solving functionalities as parts of some complex data-analysis process or for realizing intelligent assistance.[66] For information presentation, they may realize text summarization or document clustering, or similar value-adding information analysis functions.[67]
- For information acquisition, they realize user questionnaires, input interfaces, feedback gathering, critiquing components, data mining, etc.
- For knowledge maintenance, user interfaces for giving explicit feedback are possible, as well as sophisticated learning algorithms for adapting the OMIS background knowledge base.

A look into Information Agents.

What are the ontologies good for?

So far, we considered information agents pretty superficially without a closer look in their internal functioning. As far as we discussed our architecture now, we could already subsume architectures like [Klamma, 2000]'s, but there is no justification yet for our heavy-weight, ontology-based metadata approach on top of ontological representations. Though it is not a definitional element (nobody forbids to build a "simple" OMIS), we would consider it reasonable that a KnowMore-like system possess at least some "intelligent" functions at the level of Info Agents. Before we can discuss what "intelligent" shall mean in this context, let us have a look at the

---

[66] In principle, our overall system approach is to leave the problem-solving competence with the end user, but provide an expressive and extensible framewor for coordinating many "small", yet useful services, which together constitue powerful support. However, the system architecture as it is, does not hinder us to plug-in arbitrarily complex software modules as information agents. So, if one wants, the OMIS can also be seen as a generic programming framework for Intelligent Assistant systems (see also Section 3.6.1). In this case, an Info Agent could be a pretty powerful module. Whether this makes sense depends on cost-benefit calculations in a concrete application scenario. One approach which could be promising is to plug-in relatively powerful data processing modules, e.g. to realize contextualized decision support, market observation, data analysis and interpretation, etc.

[67] Such an agent would be an example for a module which resides on the KBL and has no interface somewhere outside: neither to the AL nor to the KDL is required, if the input to summarize is piped through the SRI from another retrieval agent and is then forwarded to some other presentation specialist agent, e.g. for creating a graphical knowledge map out of a textual cluster result.

internal structure of an Info Agent as shown in Figure 31. Before we can talk about "intelligence", we should think about "knowledge".

Which knowledge does an info agent possess?

Figure 31 shows a retrieval agent (though we should not forget that also knowledge acquisition and learning agents might be very useful, knowledge usage will normally be the predominent use case for an OMIS) having at least three kinds of memory area:

- There is some *local memory* for intermediate results, storage of handed-over parameters, etc.

Retrieval knowledge is the most interesting knowledge type for us

- Then we have *retrieval knowledge* dealing with all such questions as: What are relevant information sources for some kind of knowledge? How can the metadata attributes be used effectively for getting optimal results (i.e., which properties are relevant for what retrieval task under which retrieval conditions?) Which other agent could be asked to produce some input required for starting a specific search? How to perform an optimal query planning in the case of complex queries with different alternatives and maybe sources with differing Quality of Service, etc. Where to find additional back-up material when interest in a certain topic has been found?

Retrieval results may be of an unsufficient quality

  This is the core knowledge area for a retrieval agent. As the most often occuring and most interesting case from the OMIS point of view, we see the case that – because of a result set which is not good wrt. some specified and measurable criteria[68] - the agent has to apply search knowledge in order to improve the degree of quality of this answer set. We will discuss this below in more detail.

Problem-solving capabilities may complement and extend retrieval capabilities

- *Problem-solving knowledge* might also be available to an Info Agent in order to fully or partially solve simple end user problems, in order to provide non-trivial services for other agents, in order to highly aggregate, analyse, or interpret found information, or for offering partial solutions to a complex decision problem.

[68] Such criteria could be, for instance: too many or too few results, results with unsufficient reliability, user not happy with result – without giving clear reasons, unsufficient coverage of a knowledge area, not detailedly enough justified decision suggestions, etc.

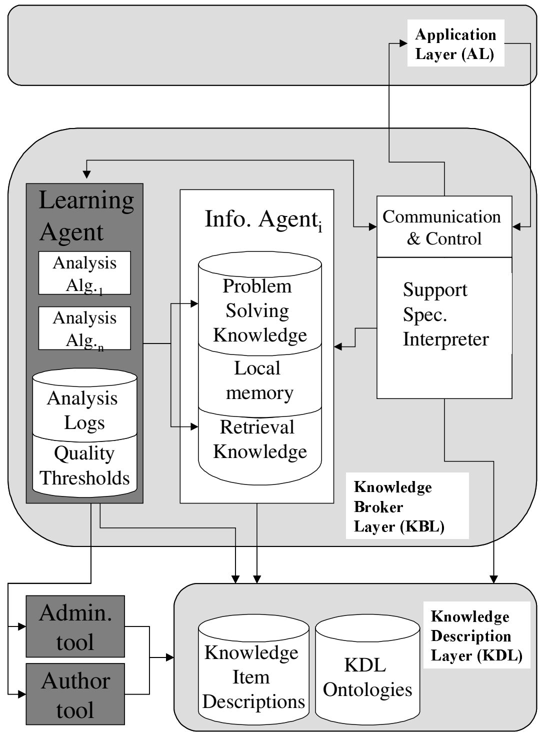


Figure 31: Generic Architecture for Learning in the KBL

With the help of such kinds of knowledge, access to the background knowledge and context data available in KDL and AL, and with formal knowledge processing methods, we can imagine at least three kinds of more intelligent information agents to populate our KBL:

Intelligent Retrieval Agents employ particularly their retrieval knowledge, reason about factual and categorical (i.e. instance and concept level) background and context knowledge - in particular query context and domain ontologies – in order to (i) better identify information sources, (ii) more precisely describe search constraints, (iii) deal with too much or too few results, (iv) deal with "wrong results", or (v) manipulate results for better further processing by other agents or for the end user. As a result, users get more precise, more relevant, and maybe better contextualized results from retrieval, query, and subscription services.

Intelligent Maintenance Agents employ particularly their problem-solving knowledge (here, a special kind: for learning system-improvements), and reason about factual and categorical context and background knowledge – especially: user behaviour, query logs, feedback information – in order to improve the overall system behaviour with respect to some measurable objectives, such as search precision and recall. As a result, retrieval knowledge of Info Agents may be adapted, as well as domain ontologies, metadata objects, or even extended process models.

Intelligent Acquisition Agents employ retrieval and problem-solving knowledge and reason about factual and categorical background and context knowledge – especially context data – for preparing knowledge storage and knowledge integration such that rich metadata descriptions with highly linked, contextualized KIDs can be inferred automatically in order to store KOs and associated KIDs (Knowledge Item Descriptions) with minimum effort for human administrators.

We don't want to go deeply into the details of such intelligent agent functionalities, since the major objective of this thesis is to provide the programming environment and the general framework, not so much the details of specific functionalities. However, we can sketch some approaches for realization of such agents and will start with a small example to give at least a clearer idea about typical approaches.

An example for ontology-based intelligent retrieval. [69]

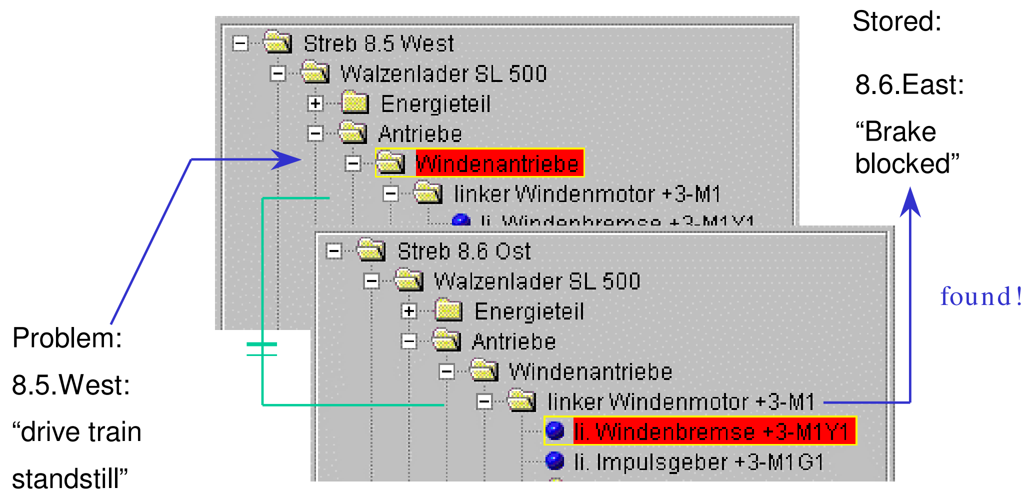


Figure 32: Example for Ontology-Based Retrieval

Figure 32 depicts a part of the search interface of the “Intelligent Fault Recording system (in German: “Elektronisches Störungsbuch - ESB”)” which is an electronic logbook for systematically recording and using maintenance experience with a highly complex technical facility in a German black-coal mine. In this system, all maintenance activities are documented as small, semi-structured text fragments which are stored with some semantic index, i.e., a link to the domain ontology, which is in this case a model of the technical facility under consideration. This model represents the part-of decomposition of the machine as well as some is-a relationships, plus the possibility to state that two installations are identical in construction. Further, electrical and hydraulic connections between system parts can be modeled.

In the example shown, the user is looking for information about machine faults or maintenance activities that happened at the mechanical drive (“Windenantriebe”)

---

[69] We use this example as an illustration which was created within the surroundings of the KnowMore project, partially using the same tools. However, the software is not the author’s original work, but mainly designed and implemented by Ansgar Bernardi, Michael Sintek, and Tino Sarodnik. Some more details on the system can be found in [Bernardi, 1997; Bernardi et al., 1998].

in the Western part of the installation (“Streb 8.5 West”). However, for this machine part, nothing has been recorded.

At that point, a simple retrieval approach would have to stop without a result.

Search heuristics rewrite an unsuccessful query

Fortunately, the ESB system allows to specify so-called “search heuristics” which describe how the background knowledge represented in the machine-model can be exploited for rewriting an unsuccessful query such that it might still lead a meaningful answer. One of these search heuristics now states that in the case of no answer which refers to a certain system X, one could look for any experience associated with some Y which is a sub-system of X (i.e. in the ontology, a part-of X). This makes sense, since often it is not intuitively clear at which level of aggregation one should enter some observation, activity, or repair. Of course, if the engine is damaged, also the car is damaged. Using this kind of knowledge, the system now extends its search to all subsystems of “Windenantrieb” in “Streb 8.5 West”. Hence, the system looks, e.g., for problems with the motor, the break, etc. Unfortunately, the answer set is still empty.

Search heuristics can be applied consecutively

However, there is some escalation mechanism for further extending the search scope by rewriting the query, by applying a second search heuristics. The second search heuristics states that maintenance entries associated with some machine part X could also be relevant for all machine parts Y – to whatever concrete installation they belong – which are identically in construction.

Search heuristics talk at the meta level about structural (part-of links between machine parts) and relational (structural-identity links) domain knowledge

With this heuristcis, the search over maintenance entries for the Western part of the installation is now extended also over entries for the Eastern part of the installation (“Streb 8.6 Ost”) which is identically constructed. Now, this two times rewritten query leads to some formation about older problem in the left brake (“li.. Windenbremse”) of this Eastern part – which gives an important hint for finding the cause of the original problem in the Western part.

Learning aspects in an OMIS.

The example above has hopefully shown that declarative formalisms for knowledge-object description and background-knowledge processing provide promising possibilities. In the same style, let us briefly illustrate which kindsof self-organization and self-adaptivity could be imagined within the KBL.

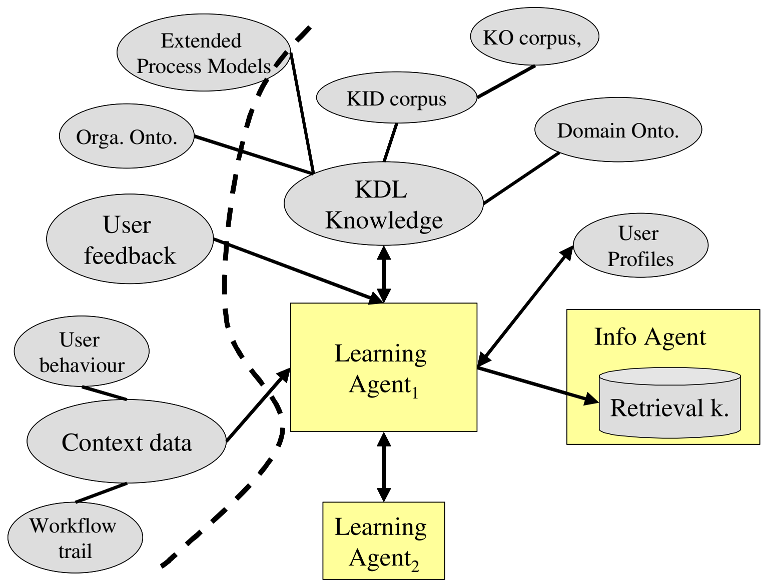


Figure 33: Kinds of Knowledge in KBL Learning

To this end, we consider Figure 33. All grey ovals represent some kind of knowledge somewhere available in the OMIS scenario. In principle, all these knowledge pieces and data can be used to improve system performance and to support self-organization. The left hand side, to the left of the dashed line, represents data and information, which will normally not be subject to change or learning, but only serve as a learning input:

Manifold inputs could be used for learning

- The whole knowledge base at the disposal of the Knowledge Description Layer: Domain Ontology and Organizational Model; Extended Business Process Models; all Knowledge Item Descriptions with their associated Knowledge Objects in the background, accesible from the KOL.
- Then we have the complete workflow instance context: technically, it is no problem to monitor complete user behaviour, search behaviour, successful and unsuccesful queries, etc.
- Further, we could have explicit or implicit user feedback. Explicit by answering question asked by the system, filling questionnaires, clicking buttons etc., implicit through interaction with the system: a link which is not followed was probably not interesting. At least the explicit feedback may fall in two categories: (a) feedback on the retrieval *process* (“this article was not relevant for that topic”), or (b) feedback on the retrieval

result, i.e. the content in the KOL ("the article fits to the topic, but its content is nonsense").

- If we had a user profiling component, it would of course be relevant, at least to be changed by learning, but maybe also to provide knowledge for certain behaviour, etc.
- Lastly, we can even imagine collaborative learning processes arising from the communication of different agents (a simple example are recommender systems).

In the recent few years, there arose a whole bunch of research literature about matters of user tracking, learning information systems, and web usage mining [Berendt et al., 2004; Oberle et al., 2003], also on exploiting semantic background knowledge or change it upon learning results [Stojanovic & Stojanovic, 2002].

Examples for learning functionalities

Our supposition is that all this work fits perfectly in our OMIS framework, because we have a maximum decoupling of concerns, a rich background knowledge, as well as a comprehensive context model. All these knowledge sources could be used, for instance, as indicated in Table 19 below.

	Input	*Affected knowledge*
Change or delete knowledge sources that are considered useless	Explicit user feedback	KOs
Explanation-based learning on unsuccessful searches	Implicit negative feedback	Info Agent's retrieval knowledge, KIDs, or KOs
Cluster typical search behaviours; analyse successful behaviour	Usage logs, implicit positive feedback; maybe user profiles and ontologies as background knowledge	Adapt retrieval knowledge of Info Agent
Usage-driven ontology evolution	Query logs, ontologies	Create, delete or rename concepts to better reflect users' domain views

Induce personalized push services	Query logs, organizational model, domain ontology	Adapt user profiles or position profiles
Induce task- or role-related push	Query logs, organizational model, domain ontology, process models and workflow instance data	Adapt info needs in extended process models
...	...	...

Table 19: Examples for Learning in the KBL

These examples may have shown that there is plenty of possibilities for such intelligent functionalities within the OMIS architecture.

Concluding remarks.

The Knowledge Broker Layer is the central instance in our architecture which offers active services and realizes intelligent functionalities. This is the reason that we put much effort in the presentation of what could be possible or in which directions one should think to evolve today's systems.

Our KBL approach is based on the Mnemonic Process metaphor

To do so, we started this Section with a thorough analysis of Klamma's framework for an OMIS implementation based on the metaphor of Mnemonic Processes, since this seems to be the most fundamental approach for understanding an OMIS system, in order not to oversee relevant or interesting ideas and opportunities. We analysed, extended and slightly changed his model and derived from it some basic architectural decisions for our KBL implementation. In particular, we clarified the often only vaguely discussed relationship between Business Processes, Knowledge-intensive business processes, KM processes, the KM meta process, and finally, Mnemonic Processes. We developed a framework which exhibits a maximum of flexibility and extensibility through a deep integration of interoperability of workflow enactment and KBL-internal reasoning, but with clearly defined interfaces.

The mnemonic process metaphor is semantically sound linked to business processes

We presented the conceptual background and a generic architecture for the KBL

On this basis, we were able to define the conceptual basis for the KBL and exhaustively discuss a generic implementation. As the basic working elements in this generic architecture, we identified Info Agents which realize Menmonic micro or macro processes.

We gave an outlook which shows innovative, intelligent functionalities within the KBL

In order to illustrate the power of the modular, extensible, highly declarative framework, we gave some examples for intelligents functionalities which could be realized within the KBL – that use background knowledge and context for precise-content retrieval and for learning and self-adaptation functions.

3.3 The Knowledge Description Layer

Knowledge is of two kinds: we know a subject ourselves,
or we know where we can find information upon it.
Samuel Johnson

3.3.1 Motivation

Formal versus informal representations.

Knowledge is always related to action

Knowledge is characterized by the fact that it can be operationalized, i.e. easily coined into concrete action. In a computer system, operationalizability can be achieved by a suitable formal representation. The more formal a knowledge representation is, the more easily and unambiguously it can be interpreted, by humans and by a machine.

Towards a meaning of formality

To make the point a bit clearer, let us introduce an "informal definition of formality", in broad terms following [Tautz, 2002]:

- Let us call some represented knowledge *formal* with respect to a specific system if *all* underlying assumptions about the way how to understand and interpret this kind of representation are either known to this system or explicitly stated in such a way that the system can understand and interpret them unambiguously (i.e., the specification of underlying assumptions is in turn formal with respect to the recipient).
- On the contrary, knowledge or information represented informally with respect to the recipient appears only as a set of meaningless symbols to this system.
- Hence a receptor system will be able to perform – in general terms – *understanding, using, and processing* of formal knowledge in an unambiguous manner, or – in more technical terms – will be able to do *query answering, knowledge manipulation and reasoning* over formal knowledge.
- As [Tautz, 2002] points out correctly, the concepts of formality / informality of a knowledge representation or a form of information storage are not absolute, but with respect to a specific system! For instance, a legal text might be very formal for a lawyer, but completely informal to an interpreter for the PROLOG programming language.

- Going further, we can understand *semi-formal systems* (corresponding verly closely to semi-structured data) as representations that contain formal and informal parts (again, cp. [Tautz, 2002]). Typically:
    - We have a formal structure in which knowledge units are organized, for instance, a labelled graph.
    - Whereas the knowledge units themeselves:
        - Maybe formal, e.g., executable programs in some programming language.
        - But also informal, like the comments in natural language meant to explain the code to a human user.
- If we analyse the landscape of really occuring knowledge representations in organisational Knowledge Management, the formal-informal spectrum is extremely wide, ranging from a metaphor in a Storytelling approach to KM [Denning, 2000; Cohen & Prusak, 2001] over technical drawings, up to a fully-automatic operationalization of a logical formalism by an automated inference engine. Figure 34 from [Scacchi & Valente, 1999] shows some widespread kinds of knowledge representation with today's associated processing mechanisms. Obviously, a suitable formalization leads to a higher degree of person independence of knowledge,[70] whereas fully-automatic processing is the extreme pole of person-independent knowledge.

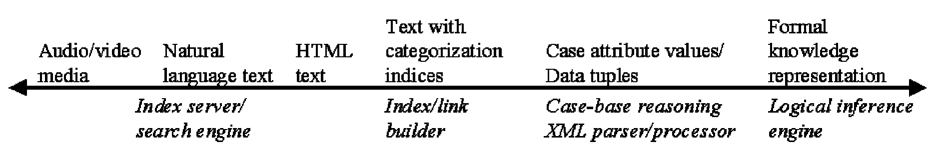


Figure 34: Representation Forms and Processing Mechanisms [Scacchi & Valente, 1999]

High degrees of formalization are often not realistic in practice

However, industrial practice shows that full-formalization is often not economically feasible or not practically possible, because system introduction and maintenance costs can become too high (cp. Subsection 1.1.5). Hence, we are looking for an economically reasonable operating point between fully formalized (like PROLOG rules) and totally informal knowledge objects (e.g. free text or drawings).

[70] which means not only the possibility to automate knowledge processing, but also easier sharing between people

The solution most widespread in KM, thoroughly employed in the area of lessons-learned systems (van Heijst et al, 1998; Weber et al., 2001), remains mostly with text documents, leaves the interpretation and case-specific usage with the human user, but technically focusses on the selection of appropriate metadata for *finding* stored lessons learned and for *assessing* their situation-specific relevance. Metadata are also a promising approach to overcome the problem of massive heterogeneity which is often inherent to a KM endeavour:

Lessons-learned systems typically rely on semi-structured text

Heterogeneity of Knowledge Objects.

Heterogeneity is ubiquitious and inevitable in an OMIS. It is "rather a feature than a bug", since it allows for cost-effective reuse of existing systems and stored content, for appropriate, people-oriented representations, and for creation of added value by linking together different forms, media, and content. As pointed out, e.g., in [Abecker et al., 1998b; Studer et al, 1999], the OMIS setting is characterized by heterogeneity in a number of different dimensions:

Heterogeneity is omnipresent

- **System level:** Typically, information relevant to be included in an OMIS, is spread over a number of different legacy and new computer systems.
  - This problem can today be overcome relatively easy by modern communication protocols (like HTTP), powerful network infrastructures, and standardized data exchange formats (e.g., on the basis of XML), or by introducing wrapper modules.

Technical distributedness is not a real problem anymore

- **Representational level:** The same content can always be represented in manifold forms, realizing different levels of formality, and stored in different media. For instance, the same knowledge could be expressed in a video or audio file with a chat over the topic, it could be written down in a free text, or also in a structured text, or it could be represented diagrammatically.
  - Hence, several early OMIS systems designed their own mechanisms for a "hardwired" way of linking together related information at different levels of formality. For instance:
    - In the KONUS design support system for mechanical engineering, design rules are stored redundantly: in an object-centered rule formalism for enactment, and in a semi-structured natural-language format for generating

The same information can be expressed and stored in manifold manners explanations for the user [Kühn & Höfling, 1994].

- In the EULE system for supporting decision processes in an insurance company, legal and company-internal regulations are expressed in a hybrid logic language for execution, but linked to original and background texts for giving understandable hints to people [Reimer et al., 1998; Reimer et al., 2000].

Communication requires a "common language", i.e. the same understanding of the basic concepts of a domain

- **Ontological heterogeneity.** Even if people are talking about the same object or domain of interest – if they have different socializations, different education, different roles in a company, a different age, nation, or religion, they may use different words for the same things, or they may use the same words for different things. Slight, but important, differences in the interpretation of technical terms – just between employees representing different departments – may be a crucial barrier for efficient communication.

Different knowledge types ask for different treatments

- **Different knowledge types and message types.** There are so many kinds of knowledge prevalent in a complex decision situation that it is already a part of the domain expertise to assess the "character" of some information and to find the right way of handling such a piece of knowledge: reliable vs. trustless, hard facts vs. rough estimations, strict rules vs. fuzzy recommendations, shallow brainstorming vs. deep thoughts, individual vote vs. broad consensus, ...
  - Normally, one would need to know relatively exact how to classify some piece of knowledge in the relevant dimensions (i.e. we need metadata attributes) and then treat it accordingly, maybe in different processes, with different procedures, in different – but mutually interacting – systems: In [Klamma, 2000], different mnemonic processes and KM workflows are defined for different kinds of knowledge in the quality improvement area.

Organizational KM typically deals with a plenty of knowledge content areas

- **Different knowledge content.** And, of course, the subject matters involved in knowledge-intensive tasks are often broad, deep, and multifaceted (cp. Section 3.4). Typically, in an enterprise one has to deal with product knowledge, technology knowledge, market and competitor knowledge, etc.
  - Here it might also be the case that for different kind of knowledge content, different representation and processing approaches are

appropriate. For instance, in our early RITA prototype for bid preparation in a complex engineering application, different kinds of productivity software (numeric analysis tools, spreadsheets, database accesses, etc.) had to be combined [Kühn & Abecker, 1997].

Are metadata the silver bullett?

Metadata shall ameliorate many of the problems above

Today's typical approach in Digital Libraries, Internet search, Web Services, ... is to attach to a Knowledge Object appropriate metadata which shall help to bridge between heterogeneous knowledge types and characters, for instance, (i) by describing how to use and manipulate from different sources and formats, (ii) by describing which ontological commitments underly a given piece of knowledge, (iii) by linking related knowledge pieces together, (iv) by characterizing relevant properties such as actuality, reliability, etc., (v) by describing the affected content in a standardized and unambiguous manner, ...

Metadata shall enable homogeneous processing of heterogeneous knowledge objects

In this way, suitable metadata shall allow to integrate *heterogeneous* and multimedia sources within the OMIS and make it accessible for Information Agents in a *homogeneous* manner. Independent from the question how some piece of knowledge is internally represented, and independent from the question to which extent this representation can be processed automatically, the metadata record shall describe clearly, ideally, in a fully machine-interpretable way, and homogeneously over all stored knowledge objects, (a) to which questions this knowledge object might contribute some value and (b) what is required for its retrieval, use and useful application.

The key is an appropriate metadata schema, we call it Information Ontology

Hence, a feasible solution for settling an OMIS upon, seems to be that we find a metadata schema for characterizing knowledge objects at a formal level, whereas the knowledge *content* itself might be represented in whatever form and media. This leads to a semi-formal data structure as introduced above (Section 0). This leads to two questions:

- How should such a metadata schema look like, i.e. which attributes do we need for characterizing knowledge objects ? (we call the answer to this question also an *Information Ontology)*; and

Besides the Information Ontology, we have to find an Ontology Language

- How we can represent and process the instances of such an Information Ontology, i.e. do we need specific knowledge representation and reasoning systems?.

In order to find an answer to these two questions, we will do an analysis of existing work, primarily in the area of Information Retrieval (IR). This will comprise the two (although pretty closely related) questions of metadata schema and of metadata language.

3.3.2 Finding the Schema: Dimensions of Information Modeling

3.3.2.1 Information Modelling in Information Retrieval

We explore the realm of logic-based Information Retrieval

The availability of almost every kind of information in electronic form, together with the success of Internet and Intranets for easy document dissemination put completely new demands on Information Retrieval technology, and theory as well. Possibly the greatest potential for facing these challenges lies in the *logic-based approach to Information Retrieval (IR)*.

Logic-Based Information Retrieval is based upon van Rijsbergen's idea to understand retrieval as the task of finding all documents **d** for a given query **q** which are likely to **imply q**, i.e., **d** $\Rightarrow$ **q** holds [Rijsbergen, 1989].

Retrieval is seen as a logical inference which can profit from different sources of background knowledge. The inference works on formal representations of both the document **d** and the query **q**. Since a user's real information need is typically specified only vaguely in the query, and, on the other hand, the content of documents can only be modeled to a certain extent, it is clear that there is a lot of vagueness and uncertainty intrinsic to this inference process. This is reflected by *probabilistic inferences* which aim at computing the probability **P(d $\Rightarrow$ q)** that d implies q.

Dimensions of document modeling in logic-based IR

Usually, document modeling in logic-based IR is concerned with three dimensions (cp. [Meghini et al., 1991; Meghini et al., 2001]):

1. the *layout structure*, e.g., of a business letter with a rectangular bold-faced region in the upper left corner of the sheet;
2. the *logical structure*, e.g., of a proceedings volume with sections as parts, articles as the sections' parts, and title, abstract, and text body as the articles' parts; and

3. the *conceptual structure*, e.g., of a technical memo which describes the content of a document, making, for instance, statements about a product's quality.

Context attributes complement document modeling

In addition to these *document-intrinsic* features, most IR systems use also some factual knowledge about the document, e.g., the author's name, the publisher etc. I will refer to these document-extrinsic features as document meta-content or document contextual structure.

Now let us discuss typical appraoches for each of these dimensions and see what they offer and what they would require from a representation language.

Document layout structure seems not of crucial importance for OMIS metadata

Layout Structure. In an OMIS, we assume that the overwhelming part of documents to be managed is available as electronic documents where layout issues are of little interest.

Moreover, automatically generated queries to the OMIS will likely not refer extensively to layout properties as manually generated queries could do which are heavily depending on the way a human user remembers documents.

Of course, layout issues have been treated in detail in knowledge-based document analysis projects (see, for instance, [Baumann et al., 1997; Bläsius et al., 1997]). It is also a vivid research topic in Multimedia Information Retrieval [Meghini et al., 2001]. Thus it would be easy to find and formulate the respective requirements for metadata attributes and representation languages. However, as said before, we will not consider this topic in the center of this thesis.

Logical document structure for document analysis

Logical Structure. Modeling logical structure of documents is also a common technique in *document analysis* [Baumann et al., 1997, Meghini et al., 1991]. There, knowledge about types of possible documents and their generic logical building blocks spans and constrains the search space for interpreting scanned documents.

Logical document structure for information retrieval

Structured-document retrieval as a branch of knowledge-based information retrieval [Rölleke & Fuhr, 1996; Fuhr, 1995; Meghini et al., 2001] deals with document structure for a number of reasons:

1. First, it allows **passage retrieval**, i.e. delivering exactly the part of a document which really contains the desired information, instead of a large

document coping with a multitude of additional, irrelevant topics. Such a more fine-grained description of documents is also the basis for combining relevance factors of document parts in order to find the most appropriate aggregation level (a paragraph, a section, or a book) to present to the user.

2. Second, the growing interest in network and **hypermedia retrieval** makes it necessary, e.g., to follow links in hypermedia documents and to appropriately propagate information about interestingness of document parts along such links. Such a mechanism is of special interest when dealing with multimedia documents which consist of aggregates of multimedia document elements.

3. Third, users may want to **query for a document's structure** they know and remember partly (e.g.: *The textbook with the phrase ``Project LILOG'' in its subtitle.*).

4. Fourth, in the presence of multiple information sources with varying media types, modeling the logical structure of information sources helps to map from conceptual structures to **access paths**. [Fuhr95c] argues that differentiating between conceptual structure and logical structure can make information retrieval more effective. [Christophides et al., 1994] present retrieval models which take into account the structure of documents and provide the possibility to query for paths which lead to the relevant part of a document.

All these objectives – understanding, categorization, and high-precision retrieval of multimedia documents – are also of utmost importance in the organizational KM setting. Hence it seems convincing to think about mechanisms for describing information source structure. In detail, the following phenomena seem interesting:

Ideal requirements for describing information structures

- **Document types:** An OMIS contains manifold types of document sources (books, memos, databases etc.) which can be arranged in an is-a hierarchy (such as: *an offer is an office letter, an invoice is an office letter,* etc.).
- **Document parts:** Complex documents are composed of simpler parts. For instance, a scientific article consists of title, authors, abstract, some sections, and references.
- **Order of parts:** Imagine a document archive where complex documents

are split into their basic building blocks (e.g., paragraphs), these elementary building blocks are directly stored in the archive, and complex documents are only represented by the links to their parts. If retrieval now evaluates a more aggregated document part as the most suitable for answering the query (e.g., a section consisting of several paragraphs), the original order of document parts must be recovered, of course.

Thinking some steps ahead, we can imagine an OMIS's document base as the knowledge server for Intranet knowledge services like personally tailorized tutorials on demand. If we want to engineer such a multimedia instructional sequence within an electronic tutorial system from building blocks like examples, figures, introductory texts etc., the order of the presentation is certainly highly relevant.

- **Links in hypermedia documents:** As a further generalization of the previous, tree-like document model, which applies to sequential, paper-based documents, hypermedia information sources introduce arbitrary links between document parts. These can be exploited for query formulation (e.g.: *Show me all web pages dealing with project descriptions which can be reached starting at the FZI homepage and following at most four navigation steps* !). Links are computationally dangerous because they may introduce cyclic relationships. Furthermore, it is not a priori clear how relevance of documents for a given topic is inherited by other documents which can be reached via a hyperlink.

Ideal requirements for content representation

Now that we have gathered some ideal requirements for modelling structure of OMIS elements, let us go to the more interesting challenge, the representation of semantic content:

Conceptual Structure. For describing document content, also called the document's *conceptual structure* by [Meghini et al., 1991], the possibilities range from pure keyword-based representations up to complete formalizations of the semantic content in some expressive knowledge representation formalism. We will briefly review this spectrum of possibilities as it is discussed in the literature. The several approaches are ordered according to increasing complexity and expressiveness of content representation:

Keyword indices

- **Keyword-based content description:** The standard approach in conventional Information Retrieval (cp. [Salton & McGill, 1983; Knorz, 1996]) represents a document as a vector of words characterising what the document talks about. Keywords can be weighted in order to reflect the importance of terms. Weights are usually produced by automated indexing techniques. The keyword vocabulary can be free or controlled, i.e., predefined in a classification system or an indexing thesaurus. Index terms are not necessarily contained explicitly in the document. Index terms may be organized, e.g., ordered by explicit dependency structures; for example, *information about the need for technology* and *the need for information technologies* could be represented as

```
(information :- need :- technology)
```

or

```
(need :- technology :- information)
```

structured keyword expressions

respectively. They can also be subdivided in **main headings** and **additional qualifiers**. Structured indexing allows to establish given relations (role indicators) between main headings and additional qualifiers which determine how to interpret relationships between them which can not be disambiguated by dependency structures (regard: *solution in water* versus *solution with water*).

Conceptual indexing

- **Concept-based content description:** With the advent of multimedia IR systems, concept-based indexing started. Here, indexing cannot rely on terms occuring in a document; instead, there must be a model of the domain of discourse such that document content can be characterised with respect to this model. Since it is nearby to use well-known domain modeling techniques and languages from knowledge-based systems to build up such a concept base, there have been considerable efforts especially in building domain models with the help of description logics. This opens possibilities for formal inferences within the domain model which support retrieval.

  The most typical example is to exploit the subsumption hierarchy to reformulate the given query if retrieval is too specific, or not specific enough, respectively [McGuinness, 1998]. One step further is done in the DEDAL system [Baudin et al., 1995] where it is allowed to explicitly formulate domain-specific search heuristics as second-order statements over the given domain model. Functionally similar is the search heuristics approach demonstrated above with the ESB example (Subsection 3.2.2).

- **Precoordinated domain concepts:** While the simple concept-based approach is essentially quite similar to keyword vector indexing–with the difference that index terms are taken from an explicit domain model which can be used for formal inferences – [Bakel et al., 1996] investigate more detailed content modeling by precoordinating index concepts (e.g.: `cures(Aspirin, Headache)`). This mimics ideas from the above mentioned structured organization of keyword indices and allows more expressive queries like, e.g.:

Example for precoordinated domain concepts

Show me all documents telling what Aspirin is good for!

```
?~:- cures(Aspirin,X)
```

or

Show me all documents concerning some remedy for heart diseases!

```
?~:- cures(X,heart_disease)
```

Composed topic index

[Schmiedel & Volle, 1996] proposed to imitate the compositionality of topic indexes of books by a similar approach in description logics introducing precoordination operators as primitive concepts and roles for semantic cases of their arguments. This allows also nested (composite) descriptions, e.g.:

Precoordinated concepts for nested descriptions

```
( Comparison
     of   ( Application
                of Description Logics
                to Configuration )
     and ( Application
                of Description Logics
                to the WWW ) )
```

Full content formalization

- **Complete formalization of document content:** There are also approaches which try to formalize document content to a larger extent. For instance, [Zarri & Azzam, 1997] proposed to translate natural-language documents into formal meta-documents which represent the semantic content in some formalized lingua franca that provides an ontology with basic templates for narrative events. These templates are instantiated by objects describing real-world ojects or events that are in turn instances of some domain ontology.

Context structures

After layout, logic, and content, we will now discuss the representation of *context* of knowledge and information items which seems to be a crucial point for OMIS applications (think back to the definitional elements of knowledge shortly presented in Subsection 1.1.6).

Contextual Structure. Under contextual structure we subsume all document meta information which is not directly contained in a document: let us list some basic categories of context-giving attributes plus some remarks whether these attributes would create additional requirements for the representation language used in the KDL.

Standard bibliographic metadata

1. Standard attributes (like author or creation date).
    - no new requirements on the representation formalism (we need only some factual assertional formalism).

Creation context

2. For documents generated within the company, their *creation context* in terms of modeled business processes and / or organizational structure might be an extremely valuable information.
    - requires attribute values which can be references to entities defined in other parts of the knowledge base (namely the enterprise ontologies).

Anchor information for further search constraints

3. [Steier et al., 1995] pointed out that besides the factors characterizing the content of searched information in a business application, knowledge delivery services have also to regard a number of *search constraints*. These concern document source and document meta information. [Steier et al., 1995] propose three categories of not content-related document meta-information, namely *form* meta-information, *quality* meta-information, and *resource* meta-information. These denote, e.g., information about medium, indexing and ease of access, expected answer time for a given query, or expected costs required to produce the answer. We see virtually all this information as properties of *information sources* rather than properties of

single document *instances*[71] or knowledge objects. Let us shortly explain those different qualifiers:

Steier's qualifiers for information sources

- **Form meta-information:** describes the kind of knowledge storage and delivery by a given source. Examples are: medium, format, indexing / ease of access, volume, access and redistribution rights, etc. Such information mainly characterizes whole information sources / services (a specific document archive, a specific database) rather than single documents. Thus it should be possible to attach such information also to sources such that it can be inherited down to single documents where appropriate.
- **Quality meta-information:** this category comprises information about how reliably a query to a given source will produce an answer. For instance, which recall and precision a query to a given source will probably produce, or what answer time is to be expected. As above, such criteria are source information and should be located appropriately in the document-source ontology.
- **Resource meta-information:** refer to the fact that in a concrete retrieval situation selection of appropriate knowledge services is a decision problem influenced by cost-benefit considerations. Skills required for using the query result (e.g., if an English speaking user gets a document written in Mandarin), time needed for answer generation, hardware requirements, operating system requirements, or software needed or some hard constraints and cost factors, respectively, to be regarded in this context. Again, this are mainly source-specific factors which may be propagated to the specific documents to be delivered by a source.

We see that for some large-scale KM project which integrates manifold knowledge sources from within and outside the organization, all these factors would soon become relevant for an optimized information logistics. For the rest of this thesis,

[71] Moreover, [Steier et al., 1995] discuss *content meta information*. This is exactly what we called *conceptual structure* before. The authors consider it *meta* information nevertheless, because a deep semantic representation of content goes well beyond the kinds of direct content representation usual in IR, which rely to the biggest extent on the actual text surface.

we can mostly ignore them nevertheless, since they don't make a difference from the academic point of view.[72] However, you will recognize some of these attributes in the INKASS example application presented in Section 5.1, where – of course – the practical application background made necessary to think about some of these factors.

In Figure 35, we shortly summarize the possible kinds of metadata seen so far, coming from Information Retrieval research. Now let us see what KM research has to offer in this area.

3.3.2.2 Information Modeling in Lessons Learned Systems and Early Work in Corporate Memories

Common approaches to OMIS organization principles [Heijst et al., 1996] reveal the following factors to be essential for determining the knowledge which is useful to support an activity:

- the *task* to be performed,
- the *role* the actor plays for this task, and
- the *domain* the task is done within.

Figure 36 and Figure 37 illustrate their approach for lessons learned characterization which was heavily driven by the CommonKADS methodology for building Knowledge-Based Systems.

[Borowsky et al., 1998] concretize these factors in enterprise terminology as *business process activity, organisational role,* and *product* to be processed.

[72] This remark is not completely true, since there are at least some requirements for the design or selection of the metadata knowledge representation formalism which come from Steier et al.'s considerations. First, we see that most of Steier's meta-information concerns information *sources* and thus should be denoted at at higher than the single-document level. Parts of this information-source metadata must be propagatable to the documents contained in a source – which in turn may overwrite some parts. Notions of uncertainty and vagueness would be a benefit. Further, document meta-information should be extensible.

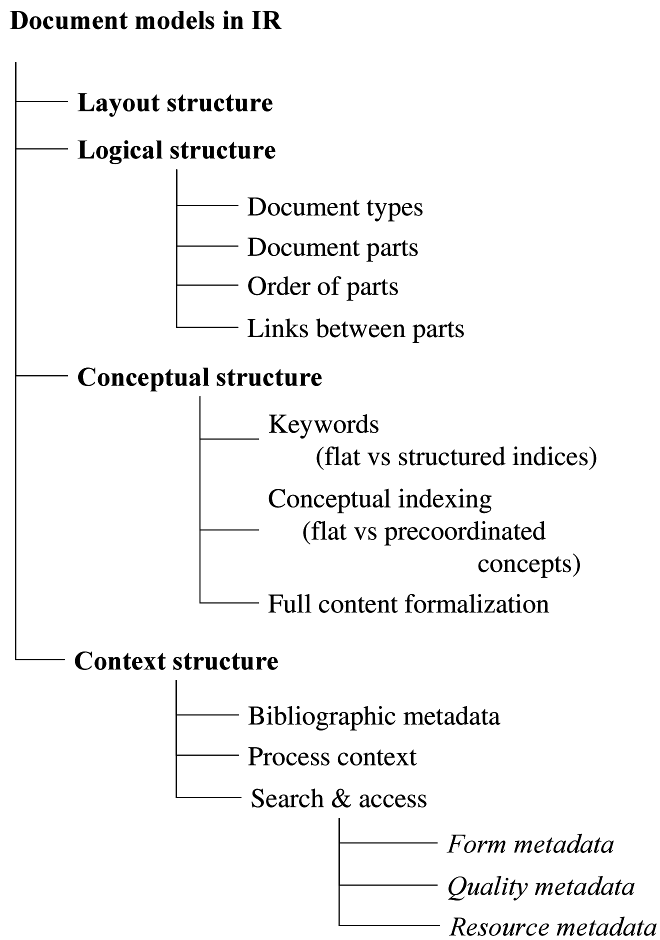


Figure 35: Overview of Metadata Types in IR Literature

This view gives us a first specialization of the general IR scenario for the enterprise knowledge management problem:

- **conceptual structure:** the topics that a knowledge object is dealing with, can be expressed in terms of the organizations' or enterprises' product models oder business objects. Of course, a useful product domain ontology will also define associated concepts like suppliers, buyers etc., and
- **contextual structure:** meta-content like the context of document-creation or possible application areas, can also be stated in terms of the enterprise ontology, the main part of which are business process models and organisational models.

General	name: role description: activity: domain(s):	 the role the knowledge is associated with the related organizational task reference to organizational areas/objects/processes
Content	generic task type: nature: products/services: functions:	from the CommonKADS library tree heuristic, formal, uncertain ... marketable products of the organization organizational functions involved
availability	time: location: form:	when available where available paper, electronic, mind, collective

Figure 36: van Heijst's seminal Information Ontology for Lessons Learned Entries

knowledge object	example
knowledge domain	internal medicine
knowledge region	urology
knowledge section	kidney diseases
knowledge segment	diagnosis of kidney diseases
knowledge element	diagnostic strategies (e.g. "first collect all symptoms, then try to explain as many of them as possible with one disease candidate")
knowledge fragment	"If the symptom is excruciating pain, then consider kidney stone"
knowledge atom	"Excruciating pain is a symptom"

Figure 37: van Heijst cont'd: Content Representation

Enterprise ontologies for Knowledge Object characterization provide for a better embedding of the OMIS in the organization

Using these formal structures for indexing knowledge items, has the advantage that already existing models, knowledge, terminology, and even formalizations can be reused and thus the OMIS is much better integrated into the organizational environment.

Compared to conventional IR approaches, we consider the context dimension very important. [Celentano et al., 1995] show how rich knowledge about business processes, started process instances, and dependencies between documents in different business process activities, can be employed for powerful search and retrieval of office letters. We adopt this view, but extend it from office letters to all knowledge and information sources used in a business process.

So far, we saw various inputs for the question how to describe organizational knowledge at a meta level. Now we will define our OMIS Knowledge Description Layer.

3.3.3 Formalizing the Knowledge Description Layer

Let us recapitulate that the advantage of such a comprehensive modeling of documents (and, of course, this holds true for any other information and knowledge source besides text documents) is the possibility to attach additional background knowledge to each of the modeled metadata dimensions and let these knowledge bases interact in the retrieval process.

Advantages of Logic-Based IR using metadata and conceptual indexing

For instance, if we have a sophisticated model of the application domain the documents talk about, we are able to index documents with pointers into this domain model. This *conceptual indexing* allows for sophisticated content representation which makes possible, e.g., formulation of domain-specific search heuristics as we saw it already in the example in Section 3.2.2. There are plenty of other examples, e.g., for the retrieval of mechanical engineering artefacts [Baudin et al., 1992; Baudin et al., 1995], for locating experts in an organization [Liao et al., 1999; Liao et al., 1999b] or – through a more precise query formulation – for better retrieval of medical information or project documents within a software organization (see, for instance, [Bakel et al., 1996]).

Moreover, conceptual indexing is a way for indexing *non-text* documents (e.g. video tapes or images, [Gordon & Domeshek, 1995] – which might play a bigger role in the future of KM – and it is a natural means for integrating information from different sources with different vocabulary (cp., e.g., [Kindermann et al., 1996]).

Justifying the ontological approach

Before we define the KDL on the basis of formal ontologies, let us briefly list the advantages that we hope to gain from this decision. Since almost no commercial system is based upon such a rich – which may mean: expensive to build and maintain – internal representation of documents or knowledge objects, as well as background knowledge, it makes sense to reflect such a decision for a moment. In practice it is even worse: not only that many systems do not maintain a declarative model of background knowledge (because they use freely chosen keywords or ad-hoc designed taxonomies for content description), some of them do not even hold a formal description of documents at all (because document content is compared with queries more or less at runtime using the text surface). And such systems are pretty widespread in practice. So let us gather some promises that characterize the ontology-based approach (some details about this can also be found in [Abecker & Elst, 2003]):

Advantages of the ontology-based approach

- Use of background knowledge for query relaxation, query refinement, etc. in order to **increase retrieval precision or recall** – as mentioned above and in the previous section.
- As mentioned above, **indexing of multimedia objects** or other entitites where it is not easy to access some text.
- Provision of **different views for different user classes** (different interest, levels of detail, wording, language, only partial views, ...) for browsing, querying, and navigation is relatively easy to be realized as mappings between presentation ontologies and stored KDL ontologies (cp. [Sintek et al., 2000]).
- Ontology **visualization** (that may use different kinds of relationships with their semantics) for improved navigation in large knowledge spaces.
- Easy support of **multilinguality**.
- Use of background knowledge for **query disambiguation**.
- Reasoning over background knowledge can detect incosnsistent queries and can be used for **explanation of search results** [Sintek et al., 2000].
- **Formal inferences** over facts of background knowledge, query and document representation, may help to close gaps between query formulation and metadata objects by inferring implicit search constraints – e.g. if I am looking about information for a project X at location Z, and I know that employee Y was one of very few people at that time at that location Z, it could be presumed that documents written by Y could refer to project X (cp. [Decker et al., 1999]).
- Ontologies serve as an excellent **target representation for Information Extraction** algorithms which distill facts and formal representations out of informal text documents, for further processing (cp. [Abecker & Elst, 2003]).
- Of course, **declarative models** are normally **easier to** understand, change and **maintain**, to some extent even (semi-)automatically (cp. [Stojanovic

& Stojanovic, 2002] about Ontology Evolution support and about usage-driven triggering of Ontology Evolution).

- The formal, logic-based semantics of ontology-based approaches allows for an **easy extension of the knowledge base** (e.g. for incorporating new types of knowledge objects, new attributes, new domain concepts), or even of the representation and reasoning paradigm (e.g. for attaching fuzzy reasoning mechanisms).
- Formal knowledge models, represented in an expressive language, are a good basis (and this was in fact the reason to create them) for **mediation between different systems** with different models.
- Formal ontologies are a good starting point for comparing complex partial models (e.g., two large query context descriptions, or two long user behaviour logs) on the basis of **similarity assessment** between structures, with background knowledge (cp., for instance [Maedche & Staab, 2002; Cimiano et al., 2004b], or [Andreasen et al., 2003] who present a result ranking using ontology-based similarity assessment).
- As we will see below, the possibility to introduce **"virtual knowledge objects"** which are created at runtime (e.g. by a DB query), which are composed from several other knowledge objects (e.g. an e-Learning course composed from different kinds of learning material), or which are just pointers to parts / paragraphs of existing documents.
- Further the possibility to explicitly introduce, and attach with attributes, and **reason about, links and relationships** between knowledge objects or part of them, for expressing discourse structure, version history, contextual relationships, etc.

What is an ontology?

For formulating and processing such conceptual models, let us roughly introduce an **ontology** as a formal, explicit specification of the conceptualization of a given domain of interest, which represents a shared understanding between a group of actors [Staab & Studer, 2003]. Though the concept of ontology represents the technical backbone of our approach, we won't go into much detail, because there is already a huge number of excellent introductions into the topic (see also [Sure, 2003]). It shall be sufficient to describe the main idea:

An ontology represents in a formally well-understood and to some extent machine-understandable language (normally, subsets of first-order mathematical logic), the basic structures underlying our understanding or communication about a certain domain of interest. These basic structures typically include:

- Classes / Concepts (sets of things) in the domain of interest

- Instances (particular things, which belong to classes) in the domain of interest

- Properties of those things

- Also concrete property values of those things (for instances)

- Relationships among those things: which relationships exist in principle?

- In particular: which relations hold between concrete instances?

- In particular, normally a subset-superset relation between concepts (is-a), and:

- a membership relation between concepts and instances (instance-of)

- Often, properties of relationships (hierarchical relationships, cardinality, domain and codomain)

- Sometimes, more integrity constraints, axioms, or inference rules which restrict the range of possible interpretations of things denoted

Ontologies and knowledge bases

In the following, we will call sets of statements regarding issues as mentioned in (1), (3), (4), (5), (6), (7) – provided it is expressed in a formal language with a well-defined semantics – a **formal ontology**. Sets of statements of type (2), (3a), (4a), (5a) are called a **Knowledge Base**.

Further, we assume an intuitive understanding of the concept *of consistency between an ontology and a knowledge base:* For instance, in a knowledge base that is consistent wrt. some ontology, the cardinality constraints and codomains of relationships must be regarded. A formal definition for the concept of consistency can be found in [Stumme et al., 2003].

Now we are ready to define the Knowledge Description Layer.

Given a formal ontology *OO* (Organization Ontology) which formalizes the concepts of Organization, Organizational Unit, Organizational Role, Organizational Position, Person from Section 0 – i.e. the Active Entities – plus the required relationships, their domains, etc.

Given a formal ontology *BPO* (Business Process Ontology) which formalizes the concepts of Activity Specification, Business Process Model, Support Request Specification, Extended Business Process Models from Section 0 – i.e. the Transformations – plus the required relationships, their domains, etc.

Let *OO* be included in *BPO*.

Let *DO* be a formal ontology which formalizes some Domain of Interest that covers the topic areas where KNOWLEDGE OBJECTS to be managed in the OMIS shall be indexed with.

Further let *OO_Model* be a KNOWLEDGE BASE which is consistent with *OO*, *BPO_Model* be a KNOWLEDGE BASE which is consistent with *BPO* and *DO_Model* be a KNOWLEDGE BASE which is consistent with *DO*.

We call *(OO_Model ∪ BPO_Model)* the *Organization Model.*

Now we define the *Information Ontology IO* as a formal ontology which:

- describes types of KNOWLEDGE OBJECTS with their RELATIONSHIPS and ATTRIBUTES that characterize concrete instances with their properties and their interrelationships

- describes INFORMATION SOURCES with their RELATIONSHIPS and ATTRIBUTES and in particular the information which types of KNOWLEDGE OBJECTS they store and how they can be accessed.

The concepts for describing KNOWLEDGE Objects:

- may contain an ATTRIBUTE which specifies the method how to retrieve it from the INFORMATION SOURCE that it is stored in

- may contain one or more context attributes which take their values from the **Organizational Model**

- may contain one or more content attributes which take their values from the *DO_Model.*

Some remarks:

- The attribute for accessing concrete knowledge objects from information sources is not mandatory since there may be knowledge objects that exist only at the description level as "virtual objects", i.e. they combine different other knowledge descriptions into a compound knowledge objects
- Context attributes (see below) refer, e.g., to departments, or to business process models, or activity specifications
- Though it may sound a bit strange, it could also be imagined that there is a knowledge description object without a content description. In particular of this is an object which exists only in the KDL for describing a link or relationship between other knowledge objects. For such a relationship it might only be interesting which KOs are linked together.

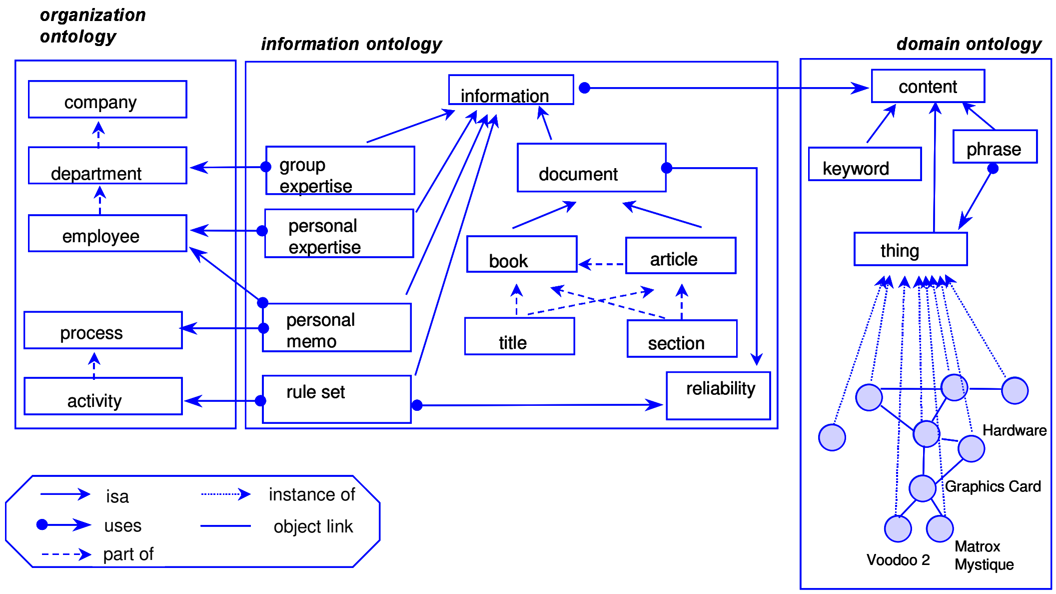


Figure 38: Most Simple Information Ontology Example

For illustrating these concepts introduced so far, let's have a look at the very simple example shown in Figure 38: There, we see in the middle of the figure a small information ontology which lists all kinds of knowledge objects to be managed in a given OMIS: documents, with the more specific concepts book and article (which may have part-of relationships to title and section parts); group and personal expertise; personal memos; and rules to be processed, e.g., with a PROLOG interpreter. Some of these concepts have a property "reliability".

At the left hand side, we see the Organization Ontology which specifies concepts such as departments, employees, or processes. At the right hand side, we have a domain ontology anbd domain model, respectively, that describes, for instance, that the Matrox Mystique is a Graphics Card.

Further, there are links between the Information and the other ontologies, because the codomain of Information Ontology attributes and relationships may be concepts from other, linked ontologies. So, a concrete instance of the concept "personal expertise" or "group expertise" might have an attribute "expertise owner" which may take as a value a concrete instance that belongs to a knowledge base instantiating the organization ontology at the right. Similarly, a personal memo might refer to a business process where it was created in (e.g., for a process improvement idea), or a rule set might be used to automate a certain task in a specific process which can be expressed by linking a suitable attribute value into the organization ontology. The same can happen for the Domain Ontology and Knowledge Base at the right hand side. For some technical document, there might be an attribute for describing its subject topic. To express this, we could point into the domain ontology.

Instances of information, domain, and organizational ontology

Now we are already at the level of instances: a *concrete* technical report (i.e. a Knowledge Object) would be represented in the system by an ontology instance of the appropriate document concept, thus creating a Knowledge Item Description (KID) or a metadata object. Some attributes or relations of such an instance would point to other objects and instances, coming from the domain knowledge base or Organizational Model, respectively (where, for example, concrete, real employees would be described as instances of the "employee" concept).

Since we saw already the genral principle we can now also show the formal definition:

> Let us call *(OO, BPO, DO, OO_Model, BPO_Model, DO_Model)* the *Background Knowledge of KDL. (KDL-Bckground)*
>
> Given an Information Ontology *IO*: We call a Knowledge Base which is consistent with *IO* the *Knowledge Item Descriptions* (*KIDs*) or *KDL-Descriptions*.
>
> Now let us define a Knowledge Description Layer as a triple:
>
> (*KDL-Descriptions, KDL-Background, KDL-Services*)
> where *KDL-Services* denotes a set of services which:
>
> - read or write (or, query with specified query search constraints) on *KDL-Descriptions* or *KDL-Background*
> - may ask for inferences or reasoning over the knowledge represented in the *KDL*

Please note: A *Knowledge Item Description (KID)* is also called a Metadata Object.

The general definition should not prescribe a concrete set of metadata ...

Hence an OMIS can be seen as a "meta information system": differently represented and formalized knowledge objects – some or many of them may be taken from legacy systems – are integrated via a common meta-level description, are equipped – if necessary – with useful links and cross-references, and are made accessible by knowledge-based retrieval algorithms

... but examples may be more concrete

The definitions above are still very open and do not yet really define how a Knowledge Object should be defined. We kept this intentionally that open, since from the ontological point of view, the different applications are so different also, that it makes no sense to define the "ultimate information ontology". However, we can give an informal overview of what we find definitely useful or required, respectively. So, we could characterize a "typical Information Ontology" as follows:

Typical elements of an Information Ontology

- It defines the description of
    - "bibliographical" metadata (e.g., author, creation date, last access date, revision history, etc.);
    - content of knowledge items (typically in terms of attributes linking into the Domain Ontology);
    - access structures, access costs (typically, attributes of an Information Source);
    - maybe contextual descriptions (creation context and potential usage context can be interesting – the attribute values link in the Organization Model);
    - maybe additional value-adding metadata (e.g., usage preconditions, usage experiences and feedback, actuality of knowledge, reliability or similar quality measures for knowledge);
    - for
        - knowledge objects (like lesson learned entries, best practices, technical reports, ideas, project reports, etc.)
        - maybe "virtual" knowledge objects which are not persistently existing but can be created dynamically at query time (e.g. by an Internet access or by data analysis procedures). Such virtual objects might be:
        - either pointers to fragments (like a specific paragraph) of an existing knowledge object; or
        - represent an aggregate combining several other knowledge objects (like a legal regulation plus related comments and

relevant precedents, or a project documentation containing bid, intermediate reports, final report)

- It allows – at the instance level – the addition of not yet represented links and relationships between knowledge objects (e.g., for representing discourse structure, version dependencies, clusters of interrelated topics, etc.)

Figure 39 below gives an idea of a maximally simple metadata object. As we mentioned already, it makes no sense to define a fixed structure for an Information Ontology. However, one may define a reference information ontology for other applications, to be taken, pruned, and particularly tailored for a new application. For an idea how such a reference Information Ontology might look like, we refer to the Knowledge Trading Scenario introduced in Section 5.1. which will give an extensive example for the concepts just introduced.

Table A. Sample knowledge-item description.

Metaproperties	Name	"How to achieve good payment conditions From Borg Inc."
	Author	James T. Kirk
	Nature	Activity-related advice
	Type	Heuristic, experience-based
	Form	English text, MS Word source, version 28.0
	Source	File E:\home\experiences\ds9-12-99pn.doc
	Availability	Always
	Costs	None
Context	Creation process	Project ds9 for Starfleet Corp. in Dec. '99
	Creation activity	Price negotiation with hardware suppliers
	Creation department	Purchasing dept.
Content	Product	20 SUN Ultra
	Supplier	Borg Inc.
	Contact person	Dr. Darth Vader

Figure 39: Simple Knowledge Description Example

The question for suitable representation languages for the KDL can be considered solved if we accept ontologies as an appropriate technology, since in the ontology research realm, the matter of expressive and comfortable, application-oriented re-presentations with efficient inferences is a common and heavily worked on topic. Our literature analysis as well as our case-study experince showed that for the begin it is sufficient to start with some basic object-oriented modeling capabilities plus a clear model-theoretic semantics implemented in efficient reasoning components. For a much more detailed discussion of representation languages, the reader is referred to [Abecker et al., 2000b].

Concluding remarks.

What metadata do we need?

In this section we started with the question what metadata schema would be appropriate for describing OMIS content. To answer the question, we gave a comprehensive overview on Information Modelling in Information Retrieval and in Lessons Learned Systems. Based upon the representational requirements of these modelling approaches and on the manifold possibilities for realizing intelligent functionalities, we decided to settle the OMIS Knowledge Description Layer on formal ontologies, represented in a logic-based knowledge representation language.

Decision 1: built upon formal ontologies

Bibliographic, context, and content metadata should be represented

As a result of the literature analysis we identified three important areas of metadata attributes: (i) general, bibliographical metadata; (ii) context metadata; (iii) content representation. Further, we emphasized the usefulness of virtual knowledge objects only residing at the KDL, and of explicit, named links for contextualizing knowledge objects by linking them together. The overall approach is modular and extensible, since the domains for attribute values can be defined within own domain, process, and organization ontologies / models. This allows also reuse of existing models and systems. The most innovative facet here is the focus on context, expressed in process, task, and role models.

3.4 The Knowledge Object Layer

Denn, was man schwarz auf weiß besitzt,
kann man getrost nach Hause tragen.
Goethe, “Faust”

3.4.1 Motivation and Basic Clarifications

Tacit knowledge was in the focus of ambitious KM approaches from their begin

As already mentioned already in the introductory Chapter of this thesis, one should not ignore or underestimate the paramount importance of *tacit and implicit* knowledge in KM. However, the primary purpose and strength of Information Technology – and thus also of Organizational Memory Information Systems – is to deal efficiently and effectively with *explicit* knowledge in electronic, machine-processable form, i.e. in particular with “information” somehow represented in the computer system. We don’t want to enter a terminological and philosophical debate about what knowledge is, compared to information, and whether electronic *knowledge* representation and processing is possible at all. Though being aware the fact that, in principle, knowledge can only exist in the heads of people, we nevertheless deal in our appoach exclusively with artefacts and representations which can be stored and manipulated by computers.

Nevertheless, information systems have their place in KM infrastructure

To justify this approach we refer to [Drucker, 1989]:

“Knowledge is information that changes something or someone—either by becoming grounds for actions, or by making an individual (or an institution) capable of different or more effective action.”

Keeping this idea in mind, it becomes clear that *information* processing can play an important role in KM. This is also corroborated by [Nonaka & Takeuchi, 1995] in their famous “tacit vs. explicit knowledge” dichotomy:

Explicit Knowledge	**Tacit Knowledge**
o Formal and objective	o Informal and subjective
o Validated by management	o Developed through practice
o Can be articulated in formal language	o Embedded in individual experience

- stored in databases, libraries, etc.	- communicated through word-of-mouth or through informal written communications

Table 20: Explicit vs. Tacit Knowledge

Information has been a knowledge carrier or locator since centuries

Some further arguments for justifying the use of information systems for knowledge management can be formulated as follows:

- **Books** have been considered the primary tool for knowledge transfer for centuries. Of course, books contain only information. However they might be written in such a manner that there is a good chance that the reader can recontextualize that information, thus reconstituting the knowledge character of the book content.
- Formal, operational knowledge represented in **Expert Systems** – provided one knows and accepts the brittleness of their applicability – withstands even strong knowledge definitions, since it allows for fully-automatically solving non-trivial problems.
- Even in the case that we have really tacit, not explicable, expert skills, we can at least **make them easier accessible** by the means of information about their location, as typically stated in Yellow Page systems or Expert Directories, or technically sophisticated Expert Finder systems [Yimam-Seid & Kobsa, 2003].

Having agreed on the assumption, that explicitly representable information may play an important role in Knowledge Management, we can introduce some basic notions as follows (cp. [Mentzas et al., 2001; Mentzas et al., 2002]) :

What is a knowledge asset ?

Clarification :

A Knowledge Asset (k.. asset) is a tangible or intangible entity which creates, modifies, and further develops knowledge. Within an organization, a knowledge asset can be:

- a person, group, or network of persons,
- a part of the organization's static or dynamic structure (explicitly or implicitly regulated organizational behaviour), or
- a part of the organization's implicit or explicit culture.

Consequently, knowledge assets are the resources that organisations wish to cultivate with their KM approaches in order to fully exploit, continuously improve, and further extend their organizational knowledge base. The more explicit this knowledge base is, the more useful can information systems be. Hence we consider next the concept of knowledge objects.

Knowledge assets are the basic targets of KM initiatives

Clarification:

What is a knowledge object?

A Knowledge Object (k. object) explicitly represents the information required to be processed (typically) by humans for being transformed into knowledge.

Knowledge derives from the information contained in Knowledge Objects through *knowledge-creating activities* that (normally) take place within and between humans.

Knowledge objects are created, modified, stored, and / or disseminated by knowledge assets.

Referring to the (semi-)formal definitions of an Enterprise Ontology in the Application Layer description (Subsection 0), we can see knowledge assets as a new sub-class of ASSETS already mentioned there, which typically (but not necessarily) belong to an ORGANIZATIONAL UNIT, and might define themselves through reference to a BUSINESS PROCESS (in the case of a community of practice).

Inserting these concepts into the Enterprise Ontology discussed before

According to [Davenport & Prusak, 1998] the above mentioned *knowledge-creating activities* include:

Knowledge-creating activities

- *Comparison:*
  how does information about one situation compares to other known situations?
- *Consequences:*
  what implications does the information have for decision and actions?
- *Connections:*
  how does this bit of knowledge relate to others?
- *Conversation:*
  what do other people think about this information?

These activities constitute, belong to, or refer to mental, sense-making processes within a certain person. They are essentially dealing with information, but trying to internally build up knowledge. They are typically elements of the mental work when dealing with knowledge-intensive activities. We see also a certain relationship to Nonaka & Takeuchi's tranformation processes between tacit and explicit knowledge.

Basically, Davenport & Prusak's knowledge-creating activities can be seen a starting point. In our opinion, many of the Mnemonic Processes shown in the OMIS Knowledge Broker Layer, represent advanced forms of such knowledge-creating activities.

Information systems and knowledge-creating activities

The knowledge objects aim to facilitate and leverage such knowledge-creating activities by providing to humans the information needed. Hence the challenge for an OMIS is to facilitate such knowledge-creating activities, by (i) providing the right bits and pieces of information; (ii) linking to other information or make it easier to find these links; (iii) enable communication (synchronous or asynchronous) with other people.

Sample knowledge assets and objects

Some examples for k. assets and associated k. objects they create:

K. assets	**K. objects**
a person	• product ideas • insights / learnings • project proposals • whitepapers
a community of practice	• best practice documentation • process improvement ideas • FAQs • guidelines for newcomers
a business / working process	• best practice documentation • company standards • R&D material • lesson learned entries
a corporate vision	• new mission statement • strategic business plan
…	• …

Table 21: Exemplary Knowledge Assets and Knowledge Objects

Functions of knowledge objects

So far we can characterise k. objects as follows:

- They act as a catalyst, enabling the fusion of knowledge flows between people, with knowledge content discovery and retrieval, sometimes enabled or facilitated through technology.
- They act, amongst other things, as the primary connecting entity for all key components in a KM system (strategy, people, processes, technology, content areas), i.e. they represent the "the KM glue".
- They make possible knowledge transfer from person to person, or from information systems to persons.
- They are typically created and maintained by KM processes (such as an innovation management process, or a lessons-learned cycle with editorial processes and organizational roles).
- They are used to search, organise and disseminate knowledge content.

Know-Net investigated the interaction of knowledge objects and KM processes

These considerations led to the major methodological design decision made in our Know-Net project [Mentzas et al., 2001], illustrated in Figure 40: namely, the explicit focus on the creation of knowledge objects by KM processes, and the concentration on the questions (a) how to efficiently deal and exploit the so-created knowledge objects, and (b) how to seamlessly integrate different KM processes and KM meta processes.[73]

The pure focus on KM processes is too narrow

However, though this model clearly explains the role of explicit knowledge objects, i.e. "information items" for Knowledge Management, it nevertheless neglects a significant part of the world which must be examined, according to all our practical projects and experience: The main limitation of this Know-Net oriented Knowledge Object approach is that it focusses exclusively on "real", pure KM processes and KM artefacts, because it was motivated by the aim to clearly define what KM is, in contrast to earlier Information Management efforts. That means, it considers only KM initiatives as a completely new approach, "besides and around"

[73] A KM meta process would be the strategic KM planning and monitoring as it is shown in the outer circle of Romhardt & Probst's [Romhardt & Probst, 1997; Probst et al., 1999] KM model. An "ordinary KM process" – as compared with an operational process such as customer care, product development, handling a bid, etc. – would be, e.g., the lessons-learned process, the innavation management process, the personnel development process within the HRM department, etc.

existing work practices. The Knowledge Objects shown in Figure 40 could, for instance, be:

- sketches of new product or project ideas in an Innovation Management module for KM;
- notes on personal learnings or personal learning plans in a Personal KM development module; or
- best practice documents in a Continuous Process Improvement inititative as part of the KM strategy.

However, as soon as we would mention "operational" data, information, and documents there, which do not belong to the KM process (which is a meta process to the basic operational business process under consideration), people would start to complain why these documents are included there, though they were already existing before the KM initiative. Typically, this accusation opens another discussion whether we consider here a *Knowledge* or an *Information* Management approach.

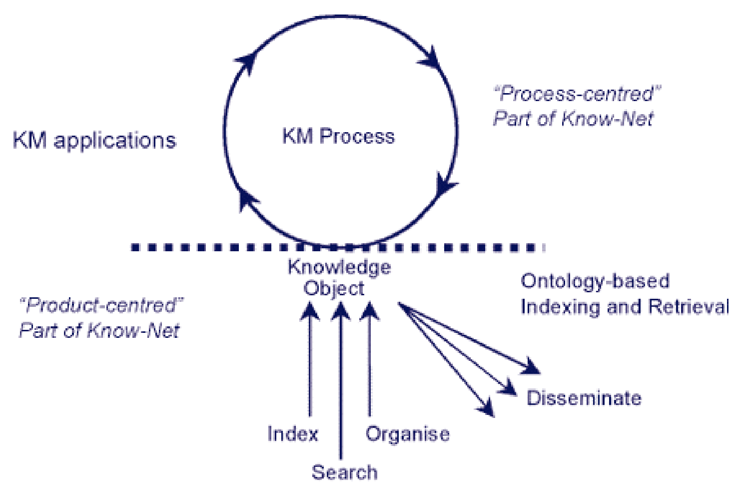


Figure 40: Knowledge Objects in the Know-Net Approach

While the resulting strong distinction between operational IM approach and meta-level KM initiative is methodologically correct and easy to understand in the historical evolution of KM theories, it imposes nevertheless severe constraints on KM approaches, thus limiting their potential impact and benefits without a clear necessity.

Our experience says that KM approaches *should not* and *cannot be* designed without a very close coupling of operational processes, documents, and information and KM process, documents, and information. There is a number of reasons for this:

KM processes must be interwoven with operational processes

- Operational data, documents, information, and artefacts of work often contain already a significant amount of knowledge which is in practice often underexploited. It is a major requirement for KM and a major source of potential KM benefits to improve the exploitation of such material. Typical examples for such documents are memos, presentations, and personal notes. The more knowledge-intensive the considered operational process is, the more important becomes this argument, usually. Often totally underexploited representations of work are, for example, technical drawings, software code, or design documents which exhibit often a big reuse potential not realized at all (cp. [Mulholland et al., 2001 ; Zdrahal et al., 2000]).
- Operational processes use (and sometimes also create) already from time to time documents and information which–also in a clear scientific analysis–would be considered KM documents and information, like, e.g., technical documentation, FAQs, document templates, or standard operating procedures. Of course, we cannot cut off these information paths already in use just for getting a scientifically sounder distinction between operational process and KM process.

If the result of this discussion is that operational and KM meta level can almost not be kept separate, it makes sense to get an idea about operational documents which typically occur in knowledge-intensive processes. Hence we will consider some examples for knowledge objects in real applications in order to get a feeling for "real-life knowledge object layers".

Example: Knowledge-Flow in the Product Lifecycle.

In Figure 41, we see a typical product lifecycle with five major phases and their several sub-activities. In Figure 42, the knowledge and information flow – in an ideal scenario – is depicted if knowledge sharing and exchange would happen optimally. Please note that Figure 42 shows only the flows which are induced by or depending on the Service activities. Some examples:

Example knowledge flows within and between phases of the product lifecycle

- Within the community of service engineers, service ideas can flow for distributing best practice through the whole company
- A relatively simple data flow exists between service and marketing/sales, production, and product development, regarding field and error data, as well as product modifications; this is really at the data level, hence it can

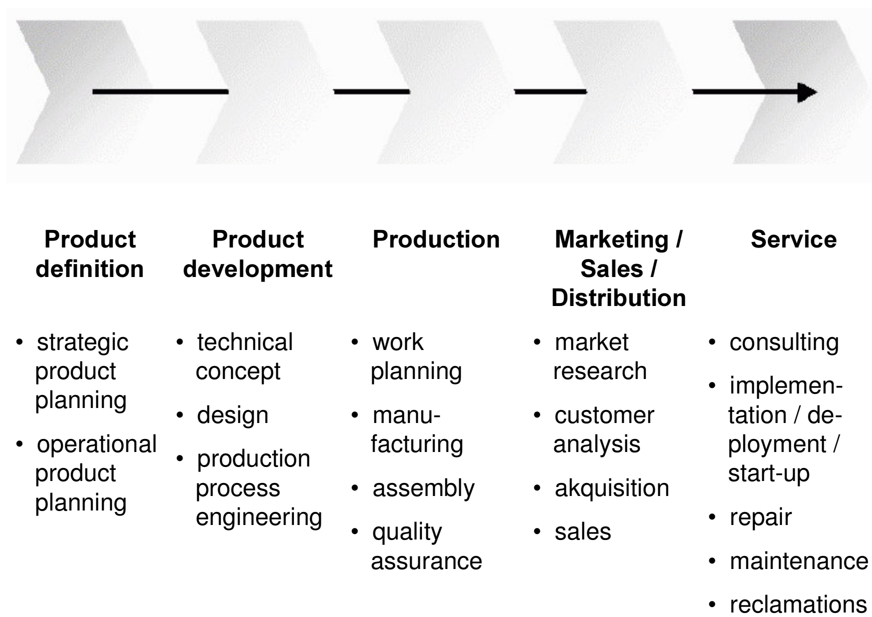


Figure 41: Product Lifecycle Steps (after [VA-3 Documentation, 1999])

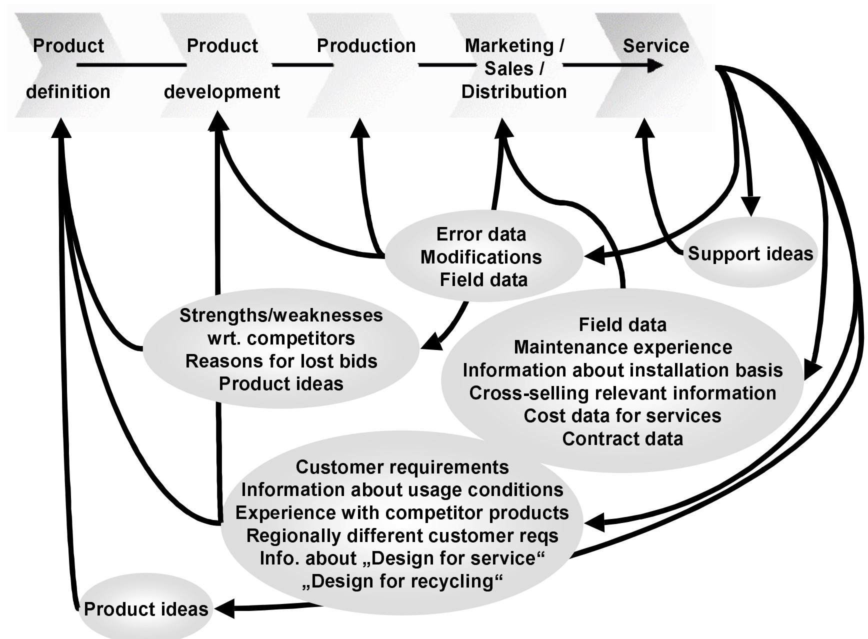


Figure 42 Information Flow Between PLC steps

be supported by simple automated methods; it maybe the case that not even a human intervention is necessary for such data exchange, but rather a machine can be tele-diagnosed. So, this maybe uninteresting for KM – except some data mining techniques would find interesting news in a bulk of such data

- A much more comprehensive set of interrelated information comes from the concrete fieldworkers working with the customer on site. They know about the field experience with a new product, learn about the concrete installation basis of a given customer – which might be worthful for detecting cross-selling opportunities, they are the best information source for determining the cost data for billing services, etc.

Service engineer are a worthful source for actual customer knowledge

  - o We can easily imagine that, normally, a service engineer is not even aware that he might possess a really worthful knowledge about cross-selling opportunities. So, one should at least give him a simple and fast connection to communicate his observations to the sales people, plus some incentive to do this.

Communication must be facilitated and stimulated

But awareness must be there before ... . and awareness comes from knowledge

- But, even worse, if he is not briefed before about new products, about the contractual situation of his customer, about the installation basis that he has to expect, he might not even *notice* that there are potentially interesting things going on at the customer site. So, we see that a massive briefing and de-briefing phase could be very useful, but also holds a huge danger of massive information overload. Here we may notice an excellent opportunity for intelligent, highly competent, task- and context-specific information sources in several phases of the work.

Compiling information into knowledge is the real art

- Climbing one level more abstract – which means more difficult to acquire, but potentially even more useful for the prior phases in the product lifecycle, such as product definition or product development – we have knowledge which comes not directly from making simple obsevations and combine them with other information [74], but really processing information, thinking about issues, aggregation of observations, clustering information and assessing their potential usefulness, ... i.e. real knowledge-intensive processing: For instance, a service engineer may discover that all customers dealing with the same materials have similar problems, or that certain times in the year are dangerous for some machines – maybe because of the climate – such information can be worthful for new product ideas, for improving quality, for sales, etc.
  - For fostering or supporting this, on one hand one could employ data mining tools as part of the KBL
  - On the other hand, it would probably be pretty useful to have a personal knowledge space for service engineers where personal notes could be taken, stored, and maybe automatically categorized, and associated with potentially interesting other information

[74] This is what we had above : hearing that a customer has some specific problem and remember that the sales department plans to roll-out a new product in this problem area within the next months.

Good ideas come from the combination of many different bits and pieces of information

This shall be enough to give an idea how complex the interaction can be between various kinds of knowledge, information, data and observations (product features and functionalities, new product ideas under work, technical and administrative data about customer installation, personal experience with similar problems / machines / customer environments, contractual and sales information and plans, experience knowledge about wear behaviour over time, maybe even local wheather data or reminders of personal partialities of business partners, ...).

DIN classification of information

After this exemplified access to the topic, let us have a look at Figure 43 which shows the [DIN 44300] classification of information according to its function. While the Action Information – "how, when, by whom" – should be encoded to a big extent into the business processes[75], the Object and the Goal Informations, respectively, are to a big extent the content of process-oriented KM systems.

Keeping in mind these basic considerations as well as the example above – which represents a typical KM application scenario – we can write up the following findings:

- For comprehensively supporting KM and knowledge-intensive tasks, the KOL will contain a broad variety of data, information, and explicit knowledge.
- The differentiation between information and knowledge is rather a theoretical than a practical question since, normally we need both for being productive and creative, and for some objects, it is hardly to decide, maybe even context-dependent whether one should consider them information or knowledge.
- Hence the integration and seamless access of already existing database, information, and documentation systems will normally play a significant role for the design of a concrete KOL.

[75] Which does not necessarily mean that no KM activities could be beneficial with respect to this knowledge type: For realizing continuous process improvements, or for running seldom, very specific processes, we have to constantly reflect and question procedures and rules, which could be well supported by providing background knowledge, e.g., about reasons for current rules, about alternatives, etc. (cp. [Wargitsch, 1997; Wargitsch et al., 1998] for ideas about evolution about process knowledge).

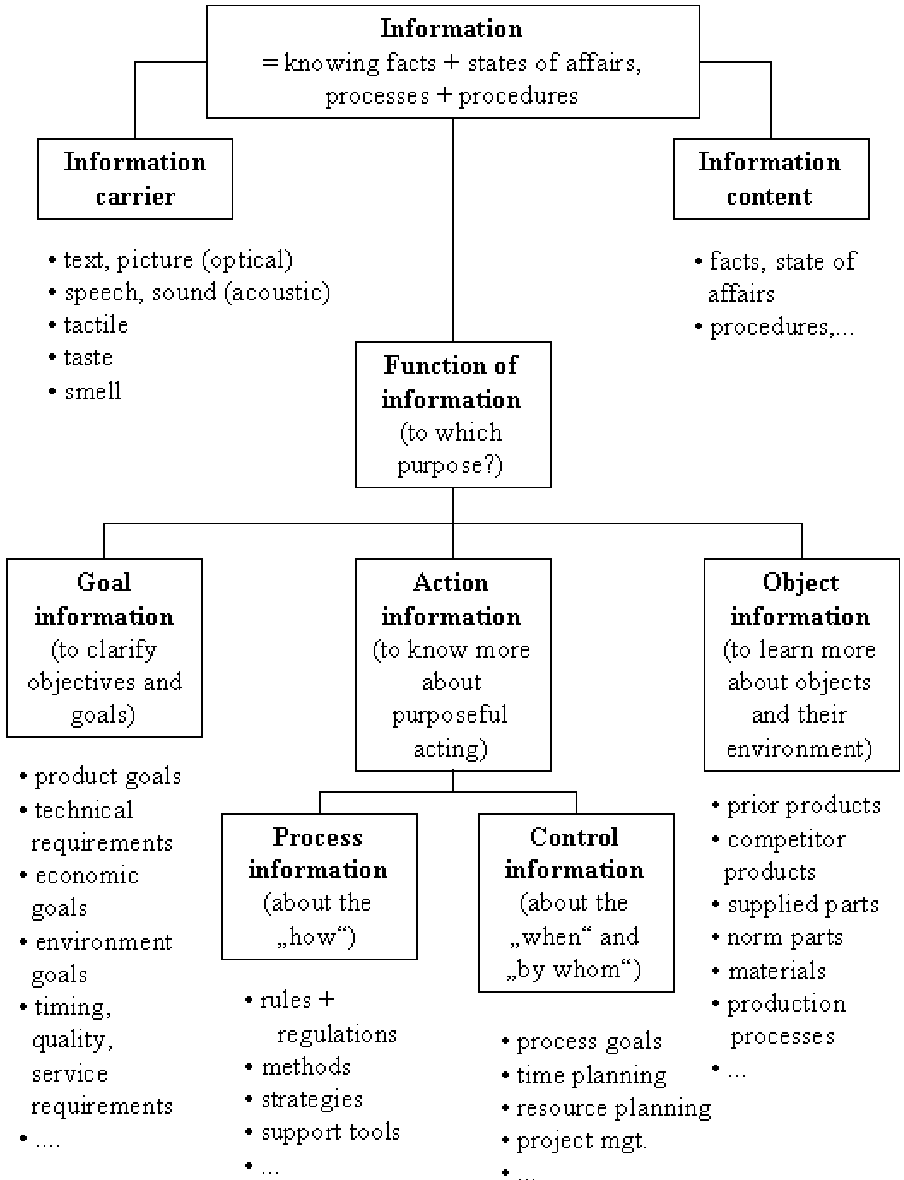


Figure 43: Information Classification According to its Function (after [DIN 44300])

- There are probably many severely underexploited document and information sources in each organization which should be systematically reviewed for their potential contributions in an OMIS scenario.[76]

---

[76] For example, in the ESB (Intelligent Fault Recording) application described above in Section 3.2.2, a significant value for the users was created by offering hyperlinks from the machine model (that was navigated anyway for inserting maintenance experience) to the respective electronic documentation and circuit diagrams

- After all this it is probably clear that the KOL must be a completely open system which allows easily to connect new information sources.
- It should also be clear that the KOL could contain information sources which are read-only, and are filled from outside the OMIS application (in the extreme case: the Internet), or otherwise, which can be filled from within the OMIS, but not only.
- When designing the KOL, it should also be investigated which information sources *could* be created newly (such as a personal notes archive, a customer-centered idea database, a cross-selling discussion group or information portal, ...) in order to foster knowledge creation and sharing, and which links should be established at the KDL in order to add value and contextualize fragmentary knowledge.

which were simply not used, before, because it was too difficult to find something in the huge documentation.

3.4.2 Formalizing the Knowledge Object Layer

Let us define:

An **INFORMATION SOURCE** IS is an ENTITY which is characterized as follows:

- It provides a SERVICE S_Read which takes as input a query expression in a query language QL_{IS} and returns as result a DATA OBJECT
- It may provide a SERVICE S_Write which takes as input a DATA OBJECT such that the following holds true:
  - if IS has performed service S_Read at least one with input DATA OBJECT DO_1 and at some later point in time, IS has to execute service S_Read with a query expression that unambiguously identifies DO_1, then DO_1 will be returned by IS

Few remarks:

Reading is mandatory ...

- Of course, real information sources may also return a set of DATA OBJECTs as query results; it would be no problem, to extend the definition such, it jsut makes it a bit more complicated, so we stay with this simple variant.

... writing is not

- A write service has not necessarily to exist. There might be information sources (e.g. an electronic version of THE BIBLE) which can be queried, but not changed.
- We adopted the service view which characterizes the functional behaviour of an Information Source, instead of, e.g., some more static data-oriented characterization which talks, e.g. about contained information elements. For us, it doesn't matter whether we consider (i) a document archive where discrete items are stored, stay unchanged, and are retrieved; (ii) a data base or a logical inference engine which stores a certain set of information, but can answer an arbitrary number of different queries, which may produce an (enumerably) infinite number of different results; (iii) a text database which might provide different services which summarize the same text at different levels of abstraction, access single paragraphs, or extract specific

information.

Now we can go on:

An **OMIS KNOWLEDGE OBJECT LAYER** KOL is a nonempty set of INFORMATION SOURCES $\{IS1, IS2, IS3, \ldots\}$.

Discussion:

- One could think about an OMIS without an KOL, i.e. with an empty set of Information sources in the KOL. This sounds not very useful to us. Of course, the several IS might be empty at some point of time, e.g.a system start-up time.
- We mentioned already earlier in this thesis that one might expect a "delete" service. This seems not to be a definitional property for us. Further, it seems not necessary and could be dangerous.
- It might make sense to define the KOL such that all writing requests from the KBL address one of the contained information sources, i.e. that there exists one IS which maybe explicitly addressed and must be able to store the type of KO which shall be stored. This would make problems if one considers a distributed scenario in a distributed company where I may be allowed to change the knowledge base of some colleague in Brasilia. This would be impossible with such a restriction. On the other hand, the standpoint makes sense that everything where I have write access belongs to my OMIS. Such that the Brasilian information source should be considered part of my memory space. We think this is not a necessary condition.
- On the other hand, if KDL is a KNOWLEDGE-DESCRIPTION LAYER for KOL, all specifications of KNOWLEDGE OBJECT access requests in the KDL must have a suited read service in one of the information sources of KOL.

Looking at the particularities of an OMIS, we make the following observations:

- Normally, an OMIS will contain ISs that are *fully under control* of the OMIS, i.e.:
    - Only the OMIS may read and write

read/write access of the OMIS to the IS may vary from source to source

- Other ISs maybe *under control* of the OMIS, i.e.:
    - The OMIS may read and write, but also others
- Some ISs may be *not under control* of the OMIS, i.e.:
    - The OMIS is allowed to read (otherwise, the IS would not be considered part of the KOL), but cannot write
        - This might be the case in read-only source.
        - Or, in a source which is under the control of some other agent.

An IS might be composed from several parts and accessed via a wrapper software

- Further, it might be the case that one logical IS for the KDL is really composed from an IS doing some query language wrapping or some result transformation.
- Another distinction criterion is the question whether there is only a unique way of querying or whether logically separate information elements can be queried separateley:
    - If there is only one query to an OMIS possible (think, that in our definition even the question for the actual could be answered by an IS), then this source is described in the KDL metadata when it occurs the first time in the system. We need only one metadata entry, even if it may return different results, which we cannot distinguish at query time. We call this a *fixed* IS.
    - But, if – and this is probably the normal case – each single information element accessible from the IS, or at least classes of such elements, must be accessed separately (e.g. via an URI), then we have to clarify when and how indexing, i.e. knowledge integration, i.e. the creation of a metadata object in the KDL, takes place. Here are two cases possible:
        - The IS *actively* notifies the KDL each time a new information element is entered such that a new metadata object can be created; then we would need a new service *Notify_KDL* which sends the query expression to the KDL which must be sent to restore the information

element.[77]

- The IS is *passive*. Then, the KDL has from time to time to find out all new elements stored in the KOL. This could be offered by another KOL-Service to be included, or by a brute-force approach of the KDL trying to get all potentially existing knowledge objects which are not yet known to the system and then integrating the not yet known returned knowledge objects.

Table 22 below shows some examples for different kinds of IS in an OMIS-KOL.

	Full control	Partial control	nocontrol	Active notification	Passive notification	Fixed IS
Commercial database service			x			x
Subscribed, external news service			x	x		
Internal Lesson Learned DB	x			N/a[78]		
Centralized LL DB shared between many partners		x			x	
Internet		(x)[79]				x
Customer DB within company		x			x	

Table 22: Types of Information Sources in the KOL

[77] Of course, in this case, we would have to extend our KDL definition by the set of such notification messages it understands.

[78] Here, as in many cases where the OMIS has write access, the question doesn't really apply: if the OMIS enters a Lesson Learned, it will have been created already the metadata object.

[79] Insofar partially under control, as one may be able to publish content over the Internet which can then be retrieved from this IS.

3.5 Problems and Limitations

Let us briefly discuss some potentially difficult aspects when introducing a software system following the principles introduced in this Chapter. The paragraphs are ordered from more concrete, technical – and thus also easier to address – issues, up to more abstract, non-technical – typically more difficult – ones. We will discuss:

Overview of potential problems

(1) Interfacing with existing systems

(2) Costs & hurdles for introducing OMIS applications

(3) Costs for creating metadata

(4) Danger to cement wrong or sub-optimal knowledge and processes

(5) Possibility of automatization of weakly-structured processes

(1) Interfacing with existing systems. Our framework requires interacting of the KnowMore "OMIS middleware" with several existing systems and services, which might produce technical integration costs that could be a barrier for introducing such approaches:

Back-end integration

Of course, the existing information systems in the Knowledge Object Layer must be accessible by the OMIS. This is today already a minor problem and will become less important with increasing standardization of protocols and access interfaces. Upcoming Intranet approaches will diminish this problem more and more. At least, many of the bigger companies have already today, or are in the process of setting-up comprehensive integration infrastructures; for instance on the basis of commercial KM tools suites. Hence this problem should not be considered critical.

Front-end integration

Not scientifically, but technically much more challenging is the Application Layer where deep integration with existing work practices is a central and important element of our whole approach. The OMIS components (i) have to monitor user behaviour, detect actions that affect decision variables or context variables, or stimulate information needs, (ii) have to present the offered information in an really "embedded", contextualized manner, and, in the ideal case (iii) have even to unintrusively capture knowledge which could be interesting for storage in the OMIS. In our KnowMore prototype implementation, we abstracted from all these technical problems by introducing the KnowMore variable editor which was fully under the control of the OMIS system. This cannot be expected to be the case in all application scenarios.

Monitoring user behaviour

Regarding the monitoring of user bevahiour for detecting actions which change decision or context variables and activate some information need, prototypical implementations of such "sniffing" software are already around for a couple of years (see, e.g., [Budzik & Hammond, 1999; Budzik & Hammond, 2000]). The more researchers and software producers will offer more intelligent user interfaces, especially with support from human-like Intelligent User Interface Agents (cp. [André et al., 1998; Rist et al., 2003]), the faster will standardization efforts in this area come forward and commercial or even Public Domain implementations become prominent. Already today interesting applications are possible without much work for dedicated special software. For instance, in their OntoOffice product, Ontoprise[80] employed the Microsoft "Smart Tag" technology for proactively offering semantics-based support to users editing Microsoft Office documents [Ontoprise, 2003]. So, we don't consider this topic critical.

Presentation of retrieval results and knowledge services

The second issue for front-end integration is how to present results and support offers of the OMIS in such a way that it will be recognized and accepted by the end user in an appropriate manner and can be integrated easily into the existing work. This question is more difficult and can probably not be expected to disappear by itself, as the latter one. Of course, today's commercial software more and more provides usable Application Programming Interfaces (API) and web-enabled user interfaces which would allow to realize functionalities such as the direct integration of a "suggested value" or of an "I – information button" into the existing applications. It is then part of the KM project management to find out whether costs and expected benefits of such "heavyweight" approaches seem to be in a reasonable balance. If not, basically there seem to be three ways to deal with this issue:

1. We keep the KnowMore approach of a separate information browser

---

[80] http://www.ontoprise.de

which – suitably designed – may act as a single point of access for all information needs an end user has. If such an information browser is designed well and incorporates both pro-active services by the OMIS and interfaces to passive information systems (typically, for instance, the Google website) this could be a simple, acceptable solution.

2. One could keep task-specific application systems and OMIS services separate, but extend the OMIS interface by a personalized User Interface Agent. This agent could try to point out task-specific information offers as clearly as possible, and try to make as easy as possible the acceptance of information offers and the integration of results into running applications.[81]

3. For a concrete application system, one could also implement an additional interface layer which encompasses functionalities of the operational tools and applications already in place, as well as new functionalities for information supply and knowledge services. Of course, this creates additional costs, but it might make much sense in concrete application scenarios, especially for increasing user acceptance. The DECOR tool suite presented in Chapter 4 already moves into this direction. There, we developed a fully integrated interface for workflow enactment, document processing, and information browsing.

Middleware integration

A last integration issue which is easy to oversee is the fact that we can expect that in many organizations there might already be workflow engines in place such that our idea of a "KM middleware" might interfere with the already existing middleware. Here we can say that our principles and methods are developed in such a way that an integration with a WfMC-compliant workflow engine should be possible with reasonable effort. It was already mentioned in Chapter 2 that many of our architectural elements can be seen as conservative extensions of the WfMC reference architecture. In [Abecker et al., 2000c], we discuss a bit more deeply the

---

[81] Such experiments were undertaken, for instance, in the Ontologging project (http://www.ontologging.com) about KM infrastructures, where human-character agents were used to point out ontology change events. However, today's pretty disappointing experiences with life-like character agents in Microsoft office applications show that there is still some work in ergonomics and Human-Computer Interfaces to do until such fancy features are widely accepted – and useful. For some ideas about application potentials of such agents in KM systems, see [Nabeth et al., 2003].

matter of integrating with existing workflow architectures.

How much effort is an OMIS introduction?

(2) Costs & hurdles for introducing OMIS applications. It is obvious that a scenario as induced by our OMIS framework will require a highly complex system design, implementation, and introduction phase, and that it might have to face manifold barriers well-known from KM introduction in general, such as the "not invented here syndrome, etc.". This is a serious problem for such technologies. Consequently, we set up the DECOR project (Delivery of Context-Sensitive Organizational Knowledge) in order to (1) develop a methodology for running OMIS introduction projects, in order to (2) implement tools for supporting such projects, and in order to (3) provide proof-of-concepts and best practice projects to demonstrate the feasibility of KnowMore-like solutions in practice. The results of the DECOR project are sketched in Chapter 4. In general, we can say that the integration of OMIS concepts with standard Business Process Management / Reengineering was very useful since the process-orientation helped much to find a focus, to guide the project, and in particular, to come to a common basis for communication with the end users. Further we could speculate that introduction efforts could decrease and success probability could increase much, if application-oriented research (or consulting companies) would come up with a reliable set of widely reusable (or, at least easy to adapt for specific new cases) *reference KM processes* (or, reference processes for widespread knowledge-intensive business processes) and / or reusable domain ontologies.

How realistic is it to build upon a strong metadata approach?

(3) Costs for creating metadata. A frequent argument against metadata-based architectures and retrieval methods is the question whether it is realistic to expect that users will spend much time for creating metadata. At least we can say that, ceteris paribus, it is indeed not realistic in today's organizations and organizational processes. This sounds a very heavy argument against our approach. Nevertheless, we think it is not. Here are some remarks answering to this question:

Maybe we could live without expressive metadata

- Even if it would really turn out that it is absolutely impossible and economically unreasonable to create suitable metadata, a significant part of our innovative contributions would still remain: At least the application-layer integration with pro-active, context-sensitive services is not critically depending from the question how stored knowledge is indexed and described. Also, the Knowledge-Description and Knowledge Brokering

Layers could be built upon weak content formalization (e.g. automatically extracted keywords and lightweight text similarity search) plus few bibliographic metadata (document type, author, creation date, ...) which are obtainable automatically. Many so-called Knowledge Management toolboxes today are working on such a basis. Then the integrative functions may degrade to some techncial source integration instead of semantic content integration (besides the possibilities that advanced text analysis methods like Latent Semantic Indexing offer).

Garbage in = garbage out

- Next, we can state that in IT – and in economics in general- often the natural law is valid: "garbage in = garbage out". Which means in this case that we cannot expect the quality and benefits of system services exceeding a certain limit if we are not willing to invest before. It is simply not reasonable to expect that IT will solve more and more semantics-based problems if we are not willing to invest in the prerequisites. It is right that today, (a) costs for building and maintaining metadata-based archives and solutions, as well as (b) benefits and return-on-investment of Knowledge Management software, are miserably understood. This makes it difficult to identify promising scenarios and exclude uninteresting ones. We expect that the future will bring more experience and thus more insights from the Business Informatics point of view in this respect.

Automatic or (semi-) automatic techniques may help

- Since the problem of metadata creation is equally important, if not more critical, in the area of Semantic Web [Berners-Lee et al., 2001] in general, there are huge research efforts since a couple of years for designing and testing fully- and semi-automatic techniques to this end. Based upon Information Extraction (see, for instance, [Kushmerick, 2000; Muslea, 2002; Muslea et al., 2003]) and Learning Text Classification [Sebastiani, 2002] technology, promising research results have been achieved both in the Semantic Web area (annotation of web-site data with semantic markup, cp. [Cimiano et al., 2004a; Handschuh & Staab, 2003]) and in the area of Organizational Memory systems (cp. [Lattner & Apitz, 2002; Lattner & Herzog, 2003]). Further, in a running, operational OMIS scenario, there will be many cases where a significant part of the metadata can be created automatically since they can be derived from the document-creation situation and context – provided document creation is done within the reach of the OMIS, and even better within an OMIS supported business process.

What happens when we enforce the widespread reuse of wrong knowledge?

(4) Danger to cement wrong or sub-optimal knowledge and processes. This is a frequently mentioned counterargument for KM approaches focussing on explicit knowledge – which is even more valid for process-oriented Knowledge Management. Here we have to say that this is first and foremost not a technical problem, but a problem of KM processes, KM structures, and KM culture. It is a clear goal of Knowledge Management to lift as much as possible knowledge in the degree of consciousness, transforming it from tacit or implicit to explicit and documented knowledge, in order to discuss and evaluate it, share and reuse it, and maybe even standardize it. Of course, this imposes restrictions on creativity and individuality, also on innovation and improvement. But it also imposes restrictions on stupidity and mediocrity. It is just a matter of careful and thoughtful system and process design: (i) to automate only processes which are suited; (ii) to restrict individual freedom only where necessary and useful; (iii) to reuse only knowledge which is worth reusing; and (iv) to build into the system possibilities to detect when changes are required. Most of these topics are issues related to the project design and process analysis, as well as issues of appropriate editorial processes for knowledge input and maintenance – including designating the required resources and installing the appropriate organizational roles for KM system administration and maintenance. Hence we do not consider this topic a counterargument for our technological approach, however, a counterargument for a frivolous use of it – which is a general theme in KM where non-technical issues turn out to be much more critical for project success or failure than the pro's and con's of technology. Nevertheless, we can, of course, make some provisions to alleviate the expected problems, like:

How to foster knowledge evolution?

- Provide easy feedback mechanism (one click, not more) in the case of useful or useless knowledge.[82]
- Provide easy possibilities to contact process owners, knowledge item

[82] This – and more feedback mechanisms – was already integrated in the KONUS system [Kühn & Höfling, 1994; Hinkelmann & Kühn, 1995; Kühn & Abecker, 1997].

owners, or information need administrators in the case of criticism.[83]

Can knowledge-work processes really be automated?

(5) Possibility of automatization of weakly-structured processes. It was already a usual argument against many workflow projects in practice, that users considered their business processes largely exposed to frequent, unforeseeable change requests due to not modeled exceptions, changing environment and requirements, etc. This argument is definitely even more legitimate if one thinks about the chaotic environment of knowledge work and the typically significant amount ad-hoc activities and hardly to automate activities like informal communications, group decision processes, or brainstormings for creative acts, etc. The situation becomes even worse if one takes into account that real "knowledge workers" typically employ very individual, often chaotic, work methods and insist in their personal sphere of freedom and autonomy. Two short remarks to this problem area:

- First, the methods presented are not necessarily useless in the presence of weak workflow structures. We expect that (a) also in relatively chaotic overall processes, some standard process templates may repeatedly occur; and (b) – vice versa – chaotic sub-sequences might be embedded in relatively clear and structured overall processes. In both cases our approach could still produce useful support provided there remains "some" context to transport dynamic workflow status into the generic information needs. Even if this is not the case, static information needs might still produce useful support. So, the argument is not necessarily a killer argument, but rather points out that OMIS requirements analysis and system introduction must be carefully planned, in order not to oversee critical human or organizational issues.
- Nevertheless, we agree that strong-structured workflow systems are a by far too restrictive tool, and some research should be directed towards more flexible and adaptive approaches. This is discussed further in Section 5.3.

[83] Such easy ways may be direct links to a chat or sending a mail or an instant message, or entering a memo in an electronic newsgroup or bulletin board. In this way the OMIS becomes a medium which fosters also human-human communication, be it synchronous or asynchronous. [Wargitsch, 1997 ; Wargitsch et al., 1998] used already such techniques to stimulate discussions for continuous business process imporvement.

3.6 Related work

Regarding other work related to our model, we consider two levels of detail: First we compare our approach with similar types of systems / research areas, namely Intelligent Assistant Systems, Electronic Performance Support Systems, and Cooperative Information Systems. Then, after this realated work "in the wider sense", we discuss a number of research implementations which are similar to our prototypes or fit roughly in our generic architectures, hence constituting related work "in the narrower sense". For those systems, we shortly summarize their major focus and contribution, sketch the differences to our approach and try to sort them into categories derivable from our generic framework.

The section discusses related work in the wider and in the narrower sense

3.6.1 Related System Classes

Intelligent Assistant Systems (IAS): They can be seen as a specific subclass or as a further development of Expert Systems. They are typically built for highly complex, time-critical and information-overloaded, real-time, decision and process-management situations. For instance, [Brézillon et al., 2000a] present a system to support subway control in the Paris metro in peak hours in the presence of incidents and unforeseen irregularities, other typical examples comprise, e.g., disaster management in natural catasprophes, process control and emergency treatment in nuclear powerplants, or military campaign planning and management.

Intelligent Assistants are a further development of Expert Systems

Typically, such systems are designed such that they aim at a balanced, cooperative problem-solving and decision-making between human and machine. This means, a major research focus is on the question which parts of the overall problem-solving could and should be automated, how the interaction with the human user should be designed, and how the complementary strengths of human and machine can be combined optimally. Hence cognitive aspects and interface matters are also a central research focus.

Decision-making and information-processing paradigm

Since the automated information processing comprises highly complex sub-tasks, and even the management of user interfaces and communication is thoroughly analysed and designed, virtually all parts of the system employ "heavy-weight"

Methods and techniques employed

methods of formal knowledge representation and processing. This comprises not only declarative AI methods for symbolic reasoning, but in many applications also complex numeric computations involving numeric simulation, analysis, or control models for parts of the physical system to be managed.

Representation of context

The transfer of knowledge from one situation to another is a central issue of IASs. Further, the transfer of highly-specific, situated knowledge from person to person (when shifts change, or when experience is transferred over time). Hence, representation of and reasoning over contexts is a central topic in IASs. For instance, [Brézillon et al., 2000a; Brézillon et al., 2000b] propose specific contextualized knowledge representations for describing situations and actions in the case of subway traffic incidents. Since most IAS researchers – like Brézillon and colleagues – develop their own formal knowledge representation languages for context representation and reasoning, they still pursue the typical "heavy AI" approach with special-purpose software and very expensive system development phases. Though there are similarities and interesting issues in the area of specialized context processing, and though the IAS people must also build domain and task ontologies for representing backgound knowledge, nevertheless our work explores another way in order to find out how far we can come on the basis of standard software as the host system and "lightweight approaches" at least for some parts of the context factors.

Evolution aspects

Because it was recognized that such methods require costly knowledge acquisition and continuous adaptation, evolution and automatic knowledge acquisition are considered an essential feature of the whole approach. Technologically, this is achieved, e.g., by relying upon Case-Based Reasoning methods [Aamodt & Nygård, 1995] which support easy adaptation of the system's knowledge base. In this way, at least a continuous adaptation of the knowledge content is possible, though there is normally little support for introducing completely new kinds of information needs, or even for adapting internal similarity functions. However, we need of course still an expensive start-up phase for building domain ontologies, defining case structures, and adjusting similarity measures.

The EPSS perspective follows a deeply integrated situated learning approach

Electronic Performance Support Systems (EPSS). Have been a trend in Business Software design from the early 1990'ties on [Gery, 1991]. It is not so much about a specific kind of system or architecture, but rather a requirements analysis and system design paradigm and set of principles. As [Cole et al., 1997] states, the ultimate goal is *just-in-time knowledge delivery*. To this end, thorough, performance-oriented workplace analyses, task analyses, and user interviews are undertaken in order to design a highly integrated, task-specific desktop which is specifically targeting at a high-performance electronic work environment. This desktop shall seamlessly integrate all existing tools for doing specific tasks, plus support tools, plus links to carefully selected background information and learning material. The whole system approach can be seen in the tradition of a deeply integrated *situated learning* philosophy (cp. [Gery, 2002]). This shall reduce start-up time for new employees and shall reduce the time for organizational changes, since new procedures, new background information, rules, regulations, etc. can be precisely brought to the point where the employee needs to know, notice, and use it.

Decision-making and information-processing paradigm

So, like in our OMIS approach – and in contrast to the IAS concepts – the user is fully responsible for doing the knowledge work and making decisions on his own, even if her or she may be supported by some specific tools for minor sub-tasks.

According to [Leighton, 2004], an EPSS may comprise the following components: (see Table 23).

Tools	**Information Base**	**Advisor**	**Learning**
Word Processing, Spreadsheet, Database	*On-line Documents, Reference Materials*	*Expert Advice and Coaching*	*Multimedia CBT and Tutorials*
Templates & Forms	*Info Databases, Case History Data*	*Context-Sensitive On-Line Help*	*Simulations and Scenarios*

Table 23: Components of an EPSS, following [Leighton, 2004]

Methods and techniques employed

Tools: Typically productivity software (word processing, spreadsheets, etc.) used with templates and forms, such as a word processing document.

Information Base: On-line reference information such as hypertext on-line help facilities, statistics databases, multimedia databases, case history databases, etc.

Advisor: For example, an interactive expert system, a cased-based reasoning system, or a coaching facility that guides a user through performing procedures and making decisions.

Learning Experiences: Computer-based-training systems (CBT), such as interactive tutorials, as well as multimedia training with simulations or scenarios.

Methods and techniques employed

We see that here, like in an OMIS, a task-specific, comprehensive selection of manifold information and knowledge sources is made in order to give full-fledged support and facilitate human learning by (and when) doing. The EPSS approach is even a bit more ambitious than a probably reasonable operating point for an OMIS would be, since the paradigm foresees considerable efforts for deeply melting all available tools and information sources into a consistent and optimized interface. This has the potential to produce highly beneficial support, but also the danger of being costly and error-prone, also not very flexible when considering individual work practices and interests.

Further, in many EPSS publications, there is a strong focus on *learning* material for training newcomers or finding into new situations, adopting new rules etc. I.e. there is a focus on knowledge coming from elsewhere to be integrated and *internalized*. We should not forget that in practice also, and particularly, the integration of operational data is important, as well as knowledge *creation* by collaboration with colleagues, and also support for knowledge *externalization* when new ideas, insights, and best practices arise.

Altogether, our impression is that EPSS pay for a highly specific and probably highly useful knowledge delivery with costly, very task-specific and pretty inflexible methods. In this respect, they come close to Expert and Intelligent Assistant Systems, with the difference that they do not employ to a big extent formal reasoning methods.

It can also be observed that the EPSS literature does not care much about sophisticated content representation, metadata or retrieval mechanisms, which are central elements of our techical approach. It seems that they consider retrieval an easy point, since they just integrate in a hardwired manner existing information sources and help facilities. This leads to the conclusion that their primary focus is rather on standard activities in an enterprise requiring a medium level of expertise, creativity, etc. There, a link to an FAQ system or to an online help is probably useful. For difficult knowledge-intensive tasks, however, located at the high-end of demands on the actors knowledge-rpocessing capabilities, the retrieval paradigm is probably insufficient, and would have been to be extended by more sophisticated methods. [Quesenbery, 2002] superficially discusses the integration of intellgent agents and electronic wizards at the user interface level. So, we see that it is difficult to make clear statement about EPSS, since they constantly integrate new technology ideas into their framework. However, in principle, we see a strong focus on workplace desin and HCI aspects, and not so much on internally sophiosticated knowledge processing and retrieval.

Representation of context

Since the EPSS paradigm foresees thorough task and workplace analyses, we assume that they acquire a relatively good understanding of relevant context factors. Hence their methods may lead to a deep and comprehensive understanding, but apparently the implementations are to some extent hardwired and ad-hoc (instead of declarative or even declaratively described on the basis of standards). For further development, we would consider more useful a thorough, general analysis of context factors and context management leading to reusable models and tools (as started, e.g. in the OMIS and in the CooPIS – see below – communities by work such as [Klemke, 2002; Goesmann, 2002], and similarly undertaken in the User Modeling area [Kobsa, 2001]).

Evolution aspects

Evolution of system content or of retrieval knowledge does not play a big role in the most EPSS literature. As [Laffey, 1995] points out, they are much more about *delivering* existing content than on content creation or evolution. Radical changes in what is presented and how, are not possible, anyway, because the interfaces are hardwired, specifically bundlign a given set of information sources. In particular, this makes it impossible to present *new sources* which were not considered when designing the system. [Laffey, 1995] envisions a system which monitors user behaviour at runtime and automatically gathers descriptions of contextualized problem-solving behaviour which can be used in the form of *cases* in the CBR paradigm. The system should be able to integrate access to new resources and to ask the user for reasons for decisions. While Laffey’s hypothetical system is still today pretty ambitious for widespread use, it shows nevertheless a clear direction towards mnemonic functions in an OMIS.

The CoopIS manifesto describes a far-reaching vision

Cooperative Information Systems (CoopIS). When the term CoopIS was coined and brought to a wider public, the "Cooperative Information Systems Manifesto" [Michelis et al., 1996] draw the picture of a far-reaching vision of next generation information systems, embedded in a thorough understanding of the organizational structures and goals, but also situated technologically in a networked world. The term "cooperative" is mostly coined because within a CoopIS, several, often widely distributed, heterogeneous, and belonging to different organizations, information systems communicate in order to produce useful results. A typical definition covering all aspects of the CoopIS approach is given by [Mylopoulos, 2003]:

- *An open, distributed information system that interoperates with other systems within an organizational context (syntactic and semantic interoperation) and contributes to the fulfillment of their mandate (cooperation).*
- *There are three facets to CIS: a distributed systems facet (openness, coordination, evolution,...), a collaborative work facet (workflows, business processes, CSCW,...), and an organizational facet (business processes, strategic objectives,...) [Agostini et al., 1998].*

Our OMIS definition can be seen as an instance of a general CoopIS approach.

Taking this comprehensive perspective, our approach to OMIS design and realization can certainly be understood as fully covered by the CoopIS definition. Hence, an OMIS represents a specific kind of Cooperative Information System which is characterized:

i. by the specific kind of business processes supported (knowledge-intensive operational processes, KM processes);

ii. hence, also by special development methods which must take into account the knowledge perspective – but can often be extensions of existing methods (as we will see in Chapter 4).

iii. by the kinds of information sources covered (specific KM-oriented information as well as information associated with knowledge-intensive processes);

iv. by the presented ontology- and metadata-based middleware located at the KDL and the KBL; and

v. a higher degree of informal media types to be treated and less structured data than in most CoopIS scenarios (see below)

vi. often, a lower degree of distributedness over organizational boundaries and thus a bit more "control" over the whole scenario than it must be supposed in a general CoopIS setting.

However, although the initial CoopIS manifesto (1) covered many aspects of system design for cooperativity and flexibility, (2) to some extent anticipated today's strong trends towards service-oriented and model-driven software architectures, and (3) ultimately envisioned a new kind of organization-aware, goal- and agent-based, change-oriented software engineering for business applications, however, the *real* work in the CoopIS area in the last ten years (this view is also confirmed by [Mylopoulos, 2003]) focussed much on the, at that time pressing, short-term practical, challenges. Coming from federated database and agent technology, and as a first step towards today's movements towards web-enabled commerce and Semantic Web ideas – primarily the question was tackled how to answer complex (database-like) queries with information to be found in the web or in databases accessible via network structures. This led to a number of challenging technical topics such as integrating data from different sources, query planning, query optimization, data quality, etc., and last, but not least, cooperative query answering: Here, the basic idea is to have cooperating system services, where, e.g., one system X knows where to find information in system Z and how to access it; another system Y knows what topic is really relevant, and finally, a system U may use Z's and Y's information for asking Z for specific data, which are then further processed. This, a bit more "down-to-earth" view is reflected in this definition of CooPIS:[84]:

Today's results are much about agent- and metadata-based approaches for gathering and integrating of data manifold data and information sources in the Web

"Networked computers which support individual or collaborative human work, and manage access to information and computing services. Computation is done concurrently over the network by cooperative database systems, expert systems, multi-agent planning systems, and other software application systems ranging from the conventional to the advanced."

As a consequence, CoopIS (or, call this view now "CoopIS in the narrow sense") are mainly not really concerned with Knowledge Management that aims at more than simple data provision, or with supporting the end user in some business decisions; rather, it is mainly a high-end approach to Information Retrieval (mainly

Decision-making and information-processing paradigm

[84] See http://www.hyperdictionary.com/ [Last access: 04/13/2004]

fact retrieval) from different sources, which might be difficult to find and access, and which might have to be integrated for further processing (see [Chu et al., 1996] as one example for many system prototypes).

Methods and techniques employed

As we have seen, the relevance of this research area for our work is not so much on the user interface level or in the application scenarios, but merely in the internal techniques for information finding and integration. There, the use of ontology-based information integration, intelligent agent technology, and metadata for resource identification has a strong tradition. So we can consider the OMIS Knowledge Broker Layer plus the underlying content and data structures in KOL and KDL a Cooperative Information System in the narrow sense which follows exactly the lines of thought developed there.

As such, some remarks can be made: First, topics like Quality of Service[85] or query planning are not the most pressing question in OMIS research, where we can expect to have a relatively small number of information sources which are to the most extent well-known and maybe even under my control (organization-internal resources, see item [vi] above). Techniques for dealing with vaguely specified queries and even with "vaguely described metadata" are more important in an OMIS, since indexing and query formulation are often non-trivial. Intelligent Information Integration is a common core concept, however, more important for the CoopIS area, were really *data* for further processing must be deeply integrated in order to process them with the same application, whereas for us it might be sufficient to find out that two document talk about roughly the same topic (which might be the easier challenge). Techniques for extracting data and metadata from text documents (Wrapper building & learning) is a crucial topic in CoopIS and in OMIS. Regarding methodological aspects, the innovative software engineering approaches developed in the CoopIS world (e.g., aiming to support Agent-Oriented [Zambonelli et al., 2003; Wooldridge & Ciancarini, 2001] or Goal-Oriented organization analysis [Bolchini & Mylopoulos, 2003; Giunchiglia et al., 2002]), might also be interesting for future research in KM. With the DECOR method, we proposed a straightforward methodological approach which works for processes in

[85] ... which is pretty important in a completely open scenario where you do not know exactly who your cooperation partners are, how they can be contacted, whether they can be reached with a good bandwidth always, and to which costs, how good their data quality is, etc.

stable environments. However, if we have to face a chaotic environment or a highly distributed overall setting, it might make sense to shift this paradigm (cp. also Section 5.2 on Agent-Mediated Knowledge Management).

Representation of context

Since the CoopIS in the narrow sense are more about how to *find* information for a clearly specified question, query and interpretation context – as we use the term in this thesis – does not play a significant role.

	IAS	EPSS	CIS	OMIS
Application scope	Very narrow, very specific, high-end applications	Very narrow, focus on learning and adaptivity; applications similar to OMIS	General purpose, data-oriented applications	High to medium application complexity, general purpose architecture
Integration with existing applications	IAS <u>is</u> the application	EPSS encompasses / replaces applications	Normally not discussed	Different scenarios, loose coupling more realistic
Information processing & decision making paradigm	Balanced, cooperative problem solving between man and machine	Human has the control	No complex problem solving, rather, algorithmic data processing, normally automatic	Human has the control
Methods & techniques	Formal knowledge processing in most parts of the system	Most interpretation is up to the user; retrieval is hardcoded in system design	Heavy to medium "intelligent" methods for data retrieval, integration, processing	Heavy-weight methods mainly focussed on information retrieval and presentation
Role of context	Cntral element for reasoning	Implicit, rather static task context	no	Explicit, dynamic context models for controlling retrieval
Evolution aspects	Automatic Case-base adaptation	Merely manual content update	Possible for evolving retrieval knowledge	Planned to be central (content and retrieval knowledge update) rarely implemented in prototypes

Table 24: Comparison of Intelligent Assistants, Electronic Performance Support, Cooperative Information Systems, and OMIS

Evolution aspects

The same remark holds true for the evolution issue. There are few contributions about learning among communities of information retrieval agents (regarding retrieval knowledge or schema integration knowledge) which are relevant to the long-term OMIS vision. The approach itself is of course desgined for scalability and extensibility, but there are normally no active elements searching for evolution needs or opportunities.

Our brief analysis is summarized in Table 24. Of course, if one has to distill into three lines which was done in 10 years research, some statements must be considered to be made from the bird's eye perspective.

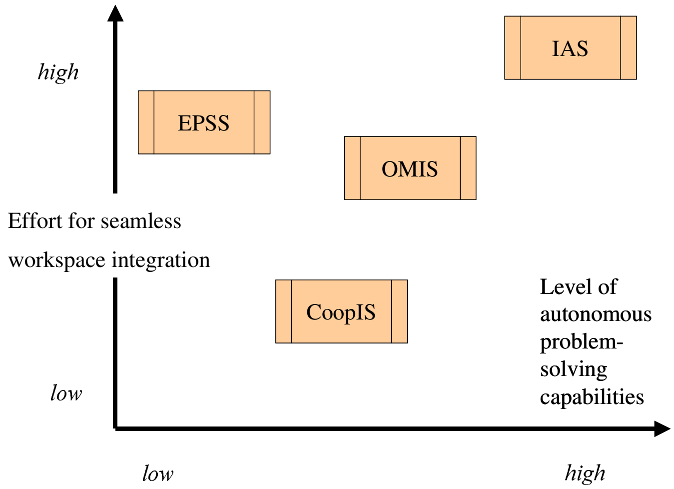


Figure 44: Rough Comparison of Related System Types

Integration depth / effort vs. support strength

Two other comparison criteria are visualized in Figure 44: here we see two central elements determining the cost-benefit ratio for each of the discussed system classes. On one hand, we consider the effort which has to be spent for integrating the daily work environment with the specific system functionalities. On the other hand, we see the potential degree of "service intelligence" offered by the system. Hence we locate Intelligent Assistants at the upper right corner: high-end system services are paid for by an expensive system set-up and case-specific design of interfaces to users and applications. EPSS deliver far less autonomous and "intelligent" system services, because the final problem-solving task stays with the user. Nevertheless, they still demand a relatively high integration effort, because they aim at a total interface integration of all affected systems, and require an expensive system analysis and introduction phase. CoopIS promise far less integration – they are just a passive information system – hence they are also "cheap" in this respect. Nevertheless, automated data analysis might offer some more specific decision-support than an EPSS does.

Finally, an OMIS is designed to occupy a new operation point between the three, where integration with existing tools should be easier – because of the business process embedding – while the range of far-reaching, autonomous services could be wider than in EPSS, because one would probably embed more problem-solving expertise into the interaction of Info Agents, context management, and process management, than in the relatively hardwired provision of learning material in an EPSS. And, of course, the CoopIS "in the narrow sense" can be considered fully subsumed by the OMIS approach, which adds a pro-active user interface, but could employ a full CoopIS realized as a set of Info Agents.

The OMIS aims at a balance between extensibility / transferability and precise, powerful support

Further, one should note that among IAS, EPSS, and OMIS, the OMIS is by far the easiest to adapt, for integrating new information sources, new processes to support, new functionalities, etc.

Altogether, the question which system type is the most appropriate – i.e. the one with the best expected Return on Investment – for a given, concrete application problem, must of course be clarified in Requirements Engineering. There are certainly processes and process types (i) which are structurally stable, and important enough that investment in a dedicated IAS or EPSS pays off.

3.6.2 Related OMIS Implementations[86]

In this subsection, we will see some other systems which followed a similar technical approach or aim at similar goals as we did with KnowMore. Hence most of these systems can be seen more or less as examples instantiating specific parts or all of the generic architecture presented in this Chapter. We present work which has primarily or exclusively been focussed on technical aspects (at least, what regards the publicly accessible information). There are few other systems that fall also in this category, but went one step further – as we did with the DECOR project – by adding methodological and modelling support. They will be discussed at the end of Chapter 4. Before we present concrete example systems, let us structure the discussion space.

We discuss systems with similar technical approach

[86] The structure of this overview goes back to the author's introductory chapter to [Abecker et al., 2002c]. Part of the presented information was gathered by Heiko Maus for joint tutorials at WM'03 and IQPC Knowledge Task 2003.

Integration Levels between OMIS and Business Processes.

As Figure 45 shows, a number of benefits can be achieved coupling an OMIS with a process management or a workflow system. The first three items below show an increasingly closer coupling and realize increasingly "smarter" information support for the user who solves a knowledge-intensive problem (the context of which is given by the workflow around).

- **Process-Oriented Knowledge Archive:** If business process models are used for organizing knowledge archives, e.g., representing one view in a company or community knowledge portal, they can be used for manual browsing. In particular, it is easy to couple an information system with the actual workflow enactment such that for a given business process activity the respective set of information objects, associated with this activity in the archive index, can be accessed. There are several commercial tools in the market realizing this idea, for instance [Fillies et al., 2001; Abecker et al., 2001].
- **Active Knowledge Services / Static Process Context:** If a workflow engine enacts a business process model, it is possible to attach information need specifications to each activity; then, the workflow system, when entering a specific activity, can automatically pose a query to the knowledge archive according to this attached information need, and proactively offer the results as information support to the user. In principle, this topic comprises all kinds of knowledge services which use an activation method by a workflow system, but do not employ an additionally modeled notion of dynamic context.
- **Active Knowledge Services / Dynamic Process Context:** As we discussed it in the previous Chapters of this thesis, if the approach above is extended in such a way that not only fixed, predefined information needs are attached to business tasks, but information needs are parameterised by variables to be filled by the running workflow instance, an even better, context-specific information retrieval can be performed taking into account instance-specific particularities.

While these three integration approaches address information searching, browsing, and access, the latter two integration ideas foster filling the knowledge archive and evolving its content during use:

- **Contextualized Information Storage:** If the concrete workflow context of a document being created is known to the OMIS at storage time, this creation context (in terms of details of the running business process instance) can be archived together with the document. This information could be used for a better retrieval in other, similar business situations, or could be used for assessing the quality of the knowledge contained (Who created it? Was the overall project successful? Is there other important background information related with this process instance? etc.). This aspect of coupling workflow and KM systems is often neglected, up to now.

- **Context-Embedded Discussions:** If a context-dependent information delivery service actively provides background information for a running business process instance, this can also stimulate discussions about content and quality of the information objects retrieved. According to the *reflection-in-action* paradigm (cp. [Sumner et al., 1999]), the user should have easy possibilities to make comments, attach discussions, send e-mails to authors or knowledge managers, etc. if a running activity gives rise to critique some information object.

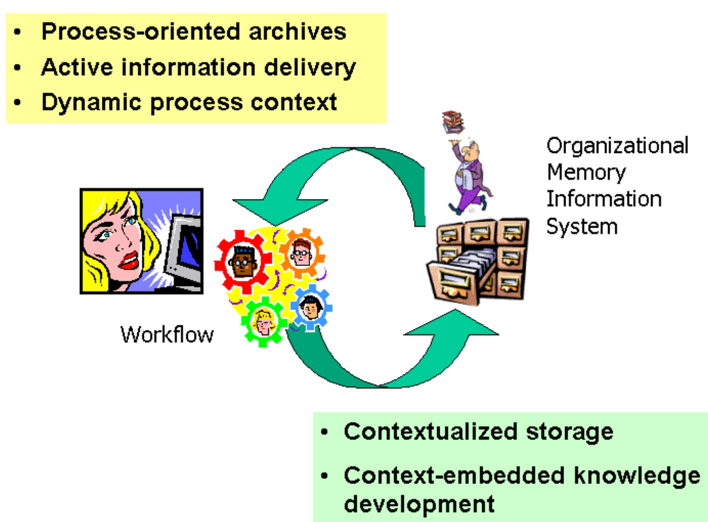


Figure 45: Different Kinds of Synergies between Workflow and OMIS

Now let us present some concrete examples: *Process portals*

DHC Vision

The **DHC Vision** Tool offered by DHC GmbH[87] is a powerful document and content management system which allows to create arbitrary semantic networks with typed structure elements and attributed links for organizing information in the enterprise [Abecker et al., 2001]. It is possible to create process models as Event-Driven Process Chains using Microsoft's graphical modelling tool Visio[88] DHC has reference applications, e.g., for knowledge organization in complex change management processes, in particular in the healthcare and life sciences area.

ARIS WebExport

The **ARIS WebExport** offered by IDS Prof. Scheer[89] follows the same idea: task and process models, organization model, information objects, etc. are modelled with the ARIS toolset for Business Process Modelling and Reengineering. The hierarchical structures of these models are then provided for browsing the models and their relationships.

ADONIS HTML export

The HTML export of the **ADONIS** Business Process Management tool suite developed by BOC GmbH[90] offers similar functionalities.

i>Knowledge-Manager

The **i>KnowledgeManager** of the ProDatO Integration Technology GmbH[91] is a commercial tool for the multidimensional indexing of a document archive. [Jablonski et al., 2002] present an example from the area of automotive engineering. Process models that have been graphically designed with the i>ProcessManager are automatically transformed into an indexing taxonomy which reflects the hierarchical decomposition of the process structure.

---

[87] http://www.dhc-gmbh.com/ [Last access: 04/17/2004]

[88] We introduce the tool here as a tool for process-oriented archives, without active, contextualized access, because this is the status of the commercially released version due April 2004. However, part of the modelling support plus a workflow extension realizing KnowMore-like services was achieved within the DECOR project presented in the next Chapter of this thesis.

[89] Now: ARIS Web Publisher™, see

http://www.ids-scheer.com/sixcms/media.php/1049/ARIS_Web_Publisher_FS_de_2004-02.pdf

[Last Access: 04/12/2004]

[90] http://www.boc-eu.com/ [Last access: 04/17/2004]

[91] http://www.prodato.de/ [Last access: 04/17/2004]

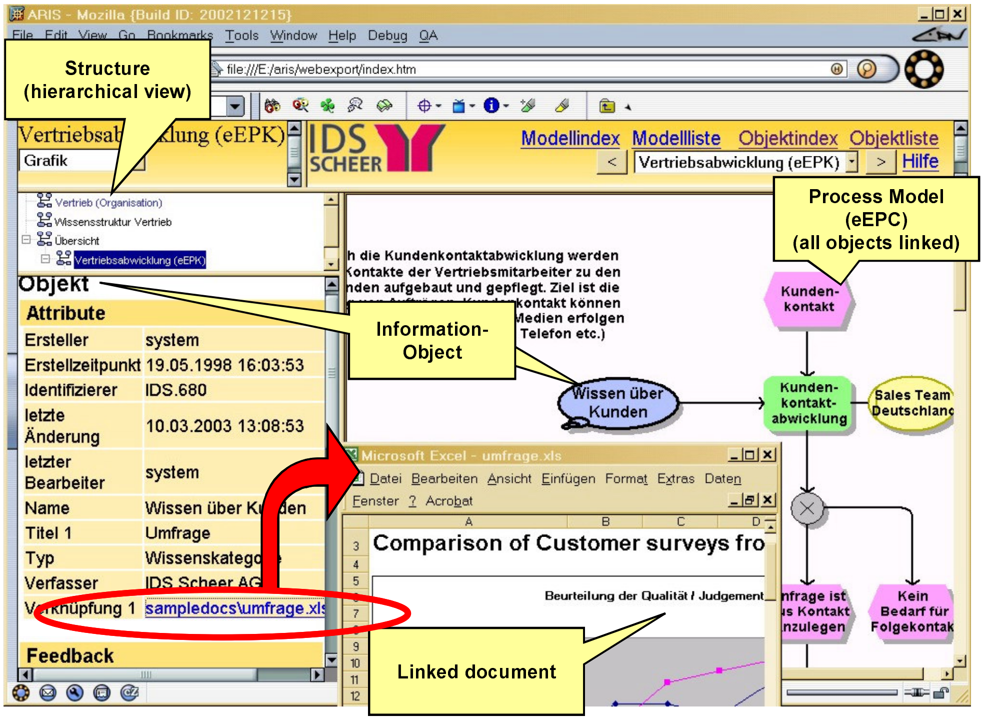


Figure 46: Example for Process-Oriented Archive: ARIS WebExport

ExperKnowledge

The **ExperKnowledge** tool provided by ExperTeam AG92 combines a Groupware and Document Management solution on top of the Livelink® tool with *process navigators*, i.e. graphically displayed process models that are hardwired with certain, related information sources, such as process-step related discussion groups [Diefenbruch et al., 2002; Brand et al., 2003]. In this way, some process-oriented structuring can be combined with a collaborative way of working in the process – using the Groupware. Since no workflow support is foreseen, the approach applies best to loosely structured processes.

POWM

The **POWM** (Project: Process-Oriented Knowledge Management to Support Collaborative Work) tool developed at FAW Ulm is a knowledge management tool for supporting engineers in the process of planning, executing and documenting ongoing engineering processes [Rupprecht 2002; Fünffinger et al., 2002]. The resulting process models can be connected with document management functionalities such that a process-oriented knowledge portal is created which dynamically evolves with the process instance.

POWM addresses ad-hoc processes

The tool is insofar very interesting, as it takes into account the high demands on process indivdualization in knowledge-intensive, collaboration-oriented work areas. It provides sophisticated support for configuring and dynamically refining process models, also – and especially – at process runtime for planning next steps.

It does not aim at process coordination like typical workflow approaches, hence it cannot deliver what we call dynamic process context for pro-active services. The focus is more on awareness, planning, and documentation of processes as they run in real-life – with a reuse of existing process elements, such as reference processes and process building blocks, plus modelling support through process-design rules which describe the influence of process-context factors on concrete process design. An interesting side effect is the evolution of process knowledge. Context-specific, active knowledge services were not in the focus of POWM, though a coupling to Information Agents has been discussed [Rose et al., 2002].

Static process context

The **KontextNavigator** developed at Dortmund University [Diefenbruch et al., 2002; Goesmann, 2002] is a prototypical add-on to the commercial CSE Workflow engine. It allows to associate documents with contexts representing processes, process steps (activities), and process instances (cases). Within a running workflow, it is then possible to access to information which has been associated with the actual working context. This is insofar a *static* context usage, as the workflow meta model is not extended for describing context parameters for information needs – as we did it. However, the process instance is recognized as a relevant context parameter such that it is, for instance, possible to access all documents related to the actual process instance.

Document analysis systems

A completely different kind of support: Document Systems like **DFKI's Office-Maid** prototype [Baumann et al., 1997], COI's **Intellidoc** product[93], or insider's **SmartFIX** product[94] [Klein & Dengel, 2004] which can be coupled with workflow. Such systems analyse scanned paper documents, in industrial applications, often business letters or forms, in order to classify them as a certain document type (an invoice, or a request for bid, ...) and to apply Information

---

[92] http://www.experteam.de [Last access: 04/17/2004]

[93] http://www.coi.com [Last access: 04/17/2004]

[94] http://www.insiders-technologies.de/ [Last access: 04/17/2004]

Extraction algorithms for creating meaningful electronic representations out of them (e.g., to extract the sum to pay out of an invoice). If those documents are stored in the OMIS-KOL, they can be categorized to responsible workflow agents, or they can start a new process instance.

Dynamic Process Context

The **TeamInformer** and **TeamFinder** prototypes of the former Siemens-DFKI Siemens Telecooperation Center (STZ) combined workflow, OMIS, and CSCW (Computer-Supported Collaborative Work) functionalities in such a way that – when a synchronuous cooperation took place, such as a teleconference – the system executed both preparatory and postprocessing functions. Using information from the workflow system, e.g., the prototype could make appointments in the participants' agendas. Exploiting or creating OMIS content, the system could in a context-sensitive manner, for example, determine the appropriate group of people to participate, prepare briefings, or store protocols and ask for debriefing documents.

OMIS + CSCW

OMIS + Document analysis

The DFKI prototype **VirtualOffice** [Baumann et al., 1997; Abecker et al., 2000; Wenzel & Maus, 2001] for knowledge-based Document Analysis and Understanding (DAU) employed dynamic task context for *expectation-generation* for the DAU modules. Using, for instance, the information that there was a request for bid sent to company XY, one can trim the DAU algorithms already such that it will be easier for them to recognize the incoming bid letter and correctly extract all relevant data.

The latter two examples were insofar interesting, as they involve new, comlex software systems into our OMIS scenario, with a mutual leverage effect for the usefulness of both. Further, both examples were also concerned with filling the OMIS automatically, which is an interesting perspective. Now coming back to some more "traditional", more reading-oriented OMIS applications.

EULE: an OMIS in an insurance company

The **EULE**, or EULE/2 system, respectively, developed by the Informatics Research group of Swisslife Corporation, was one of the earliest knowledge-based OMIS systems, realizing many of the aspects covered in this thesis (cp. [Reimer et al., 1998 ; Reimer et al., 2000]). Based on a hybrid-logics reasoning engine (description logics for domain ontology and data modeling, temporal / dynamic logics for expressing task and information flow, and deontic logics for rules and regulations governing a business process), it offered a high-level formal language for describing business processes and business cases. When working on a concrete case, the system could formally enact the logic-based descriptions of the business and legal regulations for forwarding the cases to the appropriate clerks, automatically create forms and letters, and making suggestions for the decisions to be made – as well as offering acess to the original legal or regulatory texts which lead to the suggested decisions.

Striking similarities to the KnowMore functionality

Insofar there are striking similarities to our approach: workflow enactment triggers information provision; automated tasks are combined with manual activities and suggestions for human decisions; human decisions are supported by information retrieval in the dynamic task context. Of course, since EULE was a closed, process-specific system, there were issues not covered that are or could easily be included in the modular, extensible framework that we presented in this Chapter, such as non-deductive retrieval methods (similarity, vector-space model, ...), access to external information sources, learning aspects, etc. However, comparing a framework with a concrete system instance is not really fair.

So, we can say that, seen from the functionality point of view, EULE was in many respects as capable as our systems, if not more powerful in some. However, it was still too close to traditional expert systems, difficult to understand for management, developed with proprietary, highly sophisticated AI modules, expensive to be developed, and with unpredictable maintenance costs. For such reasons, the already operational system prototype was offtaken, despite its proven usefulness!

EULE's influence on the KnowMore approach

Hence we followed with our KnowMore approach a line of research which was much more aligned with existing and accepted technologies, yet easy to extend and completed in manifold possible directions. So, one major aim of this thesis was consequently, to show how an OMIS architecture can be build in principle, integrated with existing infrastructures, starting with simple and inexpensive methods, later having all possibilities to attach additional, intelligent functionalities, integrate new information sources, cover new processes, etc.

OntoBroker / SGML nets

At **AIFB Karlsruhe**, [Staab & Schnurr, 2000; Staab & Schnurr, 2002] developed a KnowMore competitor based upon SGML nets for business-process modelling and enactment (an approach in the Petri Net line of work) and on the Ontobroker [Decker et al., 1999;] reasoner for deductive reasoning over facts and facts embedded in (over the Internet) distributed electronic documents (embedded means here, by means of semantic annotations, expressed in terms of domain ontology vocabulary known to the Ontobroker).

As for EULE, some similar remarks can be made: one one hand, the system realized very similar functionalities as KnowMore, but incorporated a very powerful and promising inference technology able to deal efficiently with background knowledge and distributed resources. On the other hand, it lacks possibilities for changing or complementing the retrieval paradigm or the kind of workflow enactment. Further it is not foreseen that a workflow agent produces knowledge, i.e. extends the OMIS KOL.

KnowWork

The **KnowWork** project at the University of Bremen is to some extent a successor to our early activities, since it was defined – among others – by early KnowMore participants [Tönshoff et al., 2001]. The project objective was primarily to bring the basic KnowMore ideas to a wider application, by building a reliable software architecture for contextualized information management, and applying it in industrial case studies, with a particular focus on cross-department and even cross-organizational aspects.

Advanced research topics in KnowWork

Consequently, the KnowWork overall aproach is not different from ours. They also defined a layer architecture similar to ours. However, at least in the publicly available literature, not much is said about (a) the application layer, and (b) the example applications. Instead, some more basic research questions were focussed on, such as ontology evolution [Sindt, 2003]; automatic metadata creation [Lattner & Herzog, 2003], or view mechanisms for ontologies [Tönshoff et al., 2002]. If such results are useful they could partially be integrated into our architecture (especially metadata creation agents). In general, KnowWork does not say much about methodological procedure (as we will do in the next Chapter).

PreBIS

The **PreBIS** (Pre-built Information Space) project coordinated by Fraunhofer IAO takes up our basic assumptions and works on task- and role-oriented information logistics in organizations, on the basis of ontologies and metadata (cp. [Härtwig & Fähnrich, 2003]). Since the basic technical approach is pretty similar to ours and the implementation phase of the project will just start at the time when this thesis is being finished, we cannot expect new technical insights from PreBIS. However, a new facet which might come out of the project is the embedding of context-based information logistics and the associated methodological approaches (as discussed in the next Chapter) in an overall organization Information Engineering methodology which includes organization-wide media and information-flow analysis and design. One slight difference between our approach and PreBIS is also that we did not consider user profiling aspects, because we had to focus our project. However, we consider User Models and User Profiles highly relevant, as said in Subsection 3.1.2. On the contrary, the PreBIS approach keeps user aspects by intention out of the game, because they consider it subsumed by role and task aspects [Hoof et al., 2003]. We would not share this opinion.

Learning in Process

The project **LIP** (Learning in Process) [Schmidt, 2004; Nabeth et al., 2004] took up our concepts of ontology-based modelling of task, role, and user facets for defining a comprehensive notion of context of an information need. The project is focussed on the e-Learning context, such that one sees neither techniques or methods for maintaining a dynamic Knowledge Object and Knowledge Description Layer supplied by manifold sources, nor the combination with a general workflow enactment as we discuss it in our Application Layer. Two interesting ideas which could also be relevant for future implementations or further extensions, respectively, of our framework, are the LIP architectural design on the basis of Web Services, and the research topic of incomplete and uncertain data which comes always into play when talking about context issues.

3.7 Summary

Let us shortly review the major contributions of this Chapter.

Application Layer

- We laid the **ontological foundations for** a clear definition of **an OMIS Application Layer.** Organized by the top-level concepts of the Core Enterprise Ontology (CEO), we carefully adapted and extended the AIAI Enterprise Ontology in such a way that (a) it is more compliant with widespread concepts and terminology in Workflow and Business-Process Management, and that (b) we could express and integrate the OMIS.specific concepts such as Knowledge-Intensive Tasks, Support Specifications, Information Needs. Further, we presented a **generic system architecture** for realizing an OMIS-AL.

Knowledge Broker Layer

- Based on a thorough analysis of Klamma's framework of Mnemonic Processes, we **identified basic classes and** illustrated **many concrete examples for Menmonic Processes to be included in an OMIS Knowledge Broker Layer**. We defined the OMIS-KBL as the connector between AL and KDL, **presented a generic architecture for an OMIS-KDL**, and identified opportunities to insert more intelligent Information Agents that systematically exploit the manifold sources of interoperating, declarative knowledge sources accessible from the KBL.

Knowledge Description Layer

- Starting from an extensive literature review for the area of intelligent information retrieval, as well as early Lessons Learned systems, we decide to design the OMIS Knowledge Description Layer fully ontology-based and identified as the **basic constituents of comprehensive OMIS metadata: (a) content descriptions in terms of a Domain Ontology, (b) context descriptions in terms of an Organizational Model and Ontology, and (c) Knowledge Object Descriptions in terms of an Information Ontology** which links into the other two dimensions.

Knowledge Object Layer

- Motivated by some real-world application examples, we illustrated the manifold sources of knowledge and information to be held in an OMIS-KDL, systematically listed the different types of information sources to be found, and gave a **formal definition of the OMIS-KOL.**

Altogether, we fully formalized the concept of an Organizational Memory Information System and gave a detailed presentation of a generic implementation. This generic architecture, in particular the Knowledge Broker Layer shall not be understood as the OMIS system, but rather describe a framework which leads to a “programming platform” which is (1) highly flexible and **extensible** through the Macro Process aproach within the KBL, together with the Workflow integration at the Application Layer; and the extensible Knowledge Object Layer; it is (2) **easy to integrate** with existing systems and work practices via the Workflow mechanism at the AL and through Information Source integration; and it is (3) **highly configurable** by the definition of concrete, organization specific Information, Organization, and Domain Ontologies.

Further, we discussed the related work at two levels of abstraction. Regarding similar classes of information systems, Electronic Performance Support Systems, Intelligent Assistants, and Cooperative Information Systems in the narrower sense have been examined. Because of the above mentioned flexibility and integration facilities, we consider an OMIS a very competitive approach with similarly powerful system types, i.e. EPSS and IAS. CoopIS in the narrower sense realize only the functionality of an OMIS KBL+KDL+KOL. Regarding CoopIS in the wider sense, from the functional point of view, our OMIS framework can be seen as a typical CoopIS, focussing on informal knowledge representations. To be more concrete, as an instantiation of Wiederholds’s general wrapper-facilitator-mediator architecure for distributed information systems [Wiederhold, 1992; Wiederhold & Genesereth, 1997].

4 A Total Solution for Business-Process Oriented Knowledge Management (BPOKM)

Make everything as simple as possible, but not simpler.
Albert Einstein

Abstract: In the last two Chapters, we saw a concrete research prototype implementation, as well as a comprehensive overall analysis of concepts, structures, and functionalities for an OMIS that realizes context-aware, process-embedded, proactive information services. In order to have a practical impact with such ideas, however, we need not only a technical system approach, but rather a *solution*. This means we have to provide (1) *all complementary tools* in an industrial strength implementation to make such a system really running; we have to offer (2) a *methodology* which guides potential users through the system planning, definition, and installation phase; and we need to demonstrate (3) potential *application scenarios*, problems, benefits, and risks in order show the feasibility and better understand the chances. These challenges were taken up in the DECOR project, refined and extended results of which we show in this Chapter. The Chapter is structured as follows:

- Section 4.1 gives a rough overview of the DECOR goals and approach.
- Section 4.3 presents the DECOR modelling method and tools.
- Section 4.2 introduces the DECOR archive system for process-oriented storage and retrieval.
- Section 4.4 shortly discusses the DECOR workflow engine.
- Section 4.5 sketches two of the DECOR case studies.
- Finally, Sections 4.6 and 4.7 summarize and analyse related work.

Preamble: *By far the biggest part of the results presented in this chapter of my thesis was achieved in the run of the European RTD project DECOR (Delivery of Context-Sensitive Organizational Knowledge). With some support from Ansgar Bernardi, I designed and wrote the workplan and technical annex of the DECOR proposal – in order to make the step from a "good idea" – as delivered with KnowMore, to a "pre-product stadium" which clearly shows a practical, and also commercial, potential. I acted as the Project Director and technical manager of DECOR, thus designing, fostering, and managing the technical results of the project. Nevertheless, the whole project team contributed significantly to the success of DECOR. First and foremost, I have again to thank Prof. Grigoris Mentzas, Maria Legal, Dimitris Apostolou, and Raphael Koumeri for their professional support in getting the proposal accepted in the hard European competition for research funding. At DFKI, Tino Sarodnik did an extraordinary job in implementing the DECOR workflow engine in a professional manner. At the ICCS institute of the National Technical University of Athens, Giorgos Papavassiliou and Spiros Ntioudis worked closely together with me – and also with the case study partners – for implementing and testing methods and tools for modeling processes and ontologies in DECOR. The results of this work represent the backbone of Subsection 4.3. Preliminary versions and simplified presentations have already been published in [Papavassiliou et al., 2002; Papavassiliou et al., 2002c; Papavassiliou et al., 2003; Papavassiliou et al., 2003b]. For creating the DECOR process-oriented archive system sketched in Subsection 4.2, we started with DHC's CognoVision tool which was at that time already pretty much at the leading edge of technology for expressive, concept-based document management. It was a pleasure to work with the DHC team. Let me mention explicitly Dr. Stephan Müller who was the main interface between DHC and DECOR. Some details about the functionality of the DECOR archive system were presented in [Abecker et al., 2001]. Besides these technical development partners, a number of colleagues worked on the DECOR case studies, either as employees of the case study hosts IKA, PVG, and CHUB, or as "facilitators" in SchlumbergerSema and ICCS. I thank particularly the case study partners in PVG and CHUB who supported DECOR without a project funding. In SchlumbergerSema, Daniel Haulet and Bernard Mathot mainly contributed to DECOR. Some aspects of our sample applications and some ideas about potential application areas of DECOR-like solutions have been described in [Abecker & Mentzas, 2001; Papavassiliou et al., 2002b; Abecker et al., 2003a]. Certainly, I forgot people who contributed more or less to the success of the DECOR project. All of them are listed in the final project report [Decor, 2002], as well as our EC project officers Agnes Bradier and Matteo Banti, and project reviewers Prof. Ann Macintosh and Matthew West who were always critical, but fair, competent and constructive. Short versions of the overall approach to the DECOR total solution can be found in [Abecker et al., 2002; Abecker et al., 2002b].*

4.1 Overview of the DECOR Project

4.1.1 Overall Project Objectives

Knowledge Management Systems (KMS) aim at a more efficient management of explicit knowledge prevalent in an organization in various forms. Two major hurdles for the success of such systems are:

- at the individual level, insufficient integration with established ways of working, and
- at the enterprise level, missing methodological and tool support for cost-effective introduction of OMIS approaches.

While the KnowMore framework showed a way how to address the first problem, one primary aim of this Chapter is to show how to overcome the latter one, thus enabling a widespread exploitation of state-of-the-art OMIS technologies. Operationally, this goal leads to the following sub-goals:

DECOR comprises methods, tools, and pilot applications

1. provide effective *methods* for analysing and modeling,
2. develop practical *tools* for exploiting and using, and
3. assess in three *pilot systems* the usefulness of ...

... formal business process models as a means for both defining context of OMIS contents, and for automatically linking OMIS contents directly into appropriate application situations. To achieve these overall objectives, three concrete objectives can be formulated:

DECOR objective 1: Sharing of process knowledge

Objective 1: Enable storage, sharing, and reuse of process-related explicit knowledge.

Business processes are a context-giving, structuring element for explicit knowledge that is prevalent in each organization, often even formally modeled for some purpose. Hence it makes sense to exploit the usage of business processes to organize knowledge archives. This enables an automatic, context-sensitive storage, allows for a more purposeful access, and allows to better integrate the process-oriented day-to-day work of the employee with the archive use – as we saw it in the previous Chapters. Hence the approach to achieve DECOR objective no. 1 is to build the DECOR Process-oriented Knowledge Archive, a software tool for creating structured archives of various (multimedia) knowledge and documents, organized around the notion of business processes performed in an organization.

Project objective 2: foster user-friendly OM exploitation

Objective 2: Ensure extensive, context-sensitive exploitation, and user-friendly access to Knowledge Archive content.

Having achieved DECOR objective no. 1, a user organization has at its disposal a rich stock of contextually enriched, explicit knowledge sources. However, all our practical experiences show that explicitly accessing information systems through complex query interfaces or Intranet portals, is often not accepted by knowledge workers which have to face constant time pressure and information overload. They neglect existing information sources, loose time with successful and unsuccessful searching, and are sometimes even not aware of beneficial information available. So, the DECOR objective no. 2 is to enable proactive, context-sensitive knowledge supply by the *DECOR Workflow-triggered Knowledge Delivery Toolkit*. This tool realizes a user-friendly, easy-to-use and understand workflow enactment, including a workflow-triggered activation method which automatically launches queries from the running workflow to the process-oriented knowledge archive described above.

So far, we are talking – roughly – about a re-implementation of functionalities already achieved with the KnowMore demonstrator, however, in a more practice-oriented and end-user oriented manner than this research prototype was.

Project objective 3: Create a methodological framework for Business-Process Oriented KM

Objective 3: Support knowledge-oriented analysis of organizations and processes.

Although knowledge management as well as efficiently managing business processes are widely recognized objectives for modern, service-oriented organizations, there is virtually no comprehensive and coherent, tool-supported framework for analysing an organization with respect to knowledge impact on processes and vice versa. Knowledge-oriented business analysis as a central KM objective would examine which knowledge and information sources are required or created in the organization's business processes, which information flow happens within and between knowledge intensive processes, and how process-intrinsic parameters influence information needs. Approaching knowledge-oriented business analysis is a central prerequisite for starting KM initiatives which supports knowledge diagnosis, accounting of intangible assets, and planned movement towards conscious strategic KM. Further, it builds the basis for process-oriented knowledge archives and efficient access to such archives. Consequently, the *DECOR Business Process & Knowledge Modelling Toolkit* gives methodological and technical support for integrated modeling and management of processes and knowledge. This comprises the elements:

- Representation means for modelling processes and process-embedded information needs, as well as ontologies and knowledge item descriptions.
- Modelling support for all these elements through appropriate editors and tools.
- A method that guides and accompanies all modelling activities, thus facilitating the organisational take-up.

This modelling support is the actual *extension* of the KnowMore approach.

4.1.2 Research Methodology for DECOR

The DECOR work followed some guiding principles, in particular:

Guiding principles of the DECOR work

- **Develop tool plus method:**

It is a common error of IT people to develop complex approaches and powerful tools, and leave the users alone with them. Normally, this results in a waste of money and resources without a better result than frustration. Knowledge Management (KM) is a typical example where accompanying measures in intervention areas such as organisational roles, processes, and culture are critical for the successful use of technology.

Consequently, DECOR, aimed at a total solution for business-process oriented knowledge management which (i) equips all software tools to be installed with appropriate methodological guidance about how to introduce them into an end-user environment, and (ii) vice versa, provides modelling tools for all steps in the introduction method that require sophisticated domain analysis and modelling activities.

- **Interleave development and test:**

Development of three pilot systems stands in the centre of the DECOR work

In order to produce practically relevant, yet innovative results, the DECOR project aimed at a balance between (i) test of innovative ideas in real application scenarios, (ii) technical consolidation of research approaches at the demonstrator stage, and (iii) development of really new approaches.

This means that a set of mutually complementing software and method modules were developed, tested in three pilot user test-beds, and iteratively improved during the project duration with feedback from the users. This strategy guaranteed project results relatively close to the market, with minimised failure risks.

The pilot systems are hosted by DECOR consortium members or companies closely connected to disseminator partners

Concretely, the DECOR work was organised around the development of three pilot systems in the medical and social security sector:

- One pilot was installed at IKA, the Greek Social Security Institute. The system supports the process of granting full old age pension to insured people which - as part of a normal administrative workflow - contains some central, knowledge and document intensive steps for finding a decision. These steps must be legally checkable, they are often done with uncertainty, are influenced by many legal regulations, and they are central for the correct result of the process. The DECOR pilot should improve a consistent, high quality of service for these decision steps.
- One pilot was placed at the interface between CHU Brugmann, a most important Brussels hospital and CPAS, the body of each city that has to deal with people who are in social, financial, … trouble. In the workflow of accomplishing the patient file and sending administrative and accounting data to CPAS there are often delays and wrong decisions made due to missing information, knowledge and experience (which is available in other steps of the process) which leads to heavy financial losses.
- One pilot was built for the Plasmaverarbeitungsgesellschaft (PVG) in Springe, Germany. This company, a subsidiary of the German Red Cross, deals with the acquisition, transport, storage, and processing of blood and blood plasma donors. In this highly sensitive application area, all software systems employed, and in particular the company's SAP R/3 installation, must be validated according to national and international laws and regulations. The process of making changes to this SAP R/3 system while keeping the validation status is document and knowledge-intensive and

was supported by the DECOR pilot system. [95]

4.1.2.1 Methodological Input

The DECOR pilot cases were built using the methodology developed within DECOR. Nevertheless, since the DECOR method aimed at a maximum reuse of existing, suitable input, it is to a large extent built upon two amalgamated existing methods, namely CommonKADS and IDEF5.

CommonKADS

CommonKADS is a methodology for development of knowledge-based systems (KBS), which is the result of the Esprit-II projects KADS II and KADS. Common-KADS holds the position of a a *de facto* standard for KBS development in Europe.

CommonKADS supports most aspects of a KBS development project, including project management, organisational analysis, knowledge acquisition, conceptual modelling, user interaction, system integration, and design. The original methodology is result oriented rather than process oriented. It describes KBS development from two perspectives:

- *Result perspective:* A set of models of different aspects of the KBS and its environment, that are continuously improved during a project life-cycle.
- *Project management perspective:* A risk-driven, spiral life-cycle model that can be configured into a process adapted to a particular project.

The models mentioned above include in particular: the organisation model, the task model, the agent model, the communication model, the expertise model, and the design model.

In recent years, CommonKADS has also been extensively used and documented

---

[95] A side remark regarding the relationship betweenthe DECOR project and the author of this thesis, to make clearer the original contributions made by which parties: The author of this thesis wrote all significant parts of the DECOR project workplan and managed it as the acting Project Director. This concerned in particular: Providing input for all other partners in order to understand and internalize the DECOR overall approach and objectives; making the basic decisions regarding the layout of the DECOR method; supervising the development of the DECOR workflow tool by Tino Sarodnik; coordinating synergies and mutual interrelationships between software development and case study work at different sites; facilitating the case studies, in particular the CHUB case, and partially PVG; major part of presenting and writing up results. The other responsible co-workers are mentioned at the begin of this Chapter and can be found in the DECOR Final Report.

for Business-Process Analysis and Design, apart from KBS development.

In DECOR, we used in particular the organisation, task, and agent models, as well as some project management aspects, for starting the case study development and for seeding the DECOR method for Business-Process Oriented KM.

IDEF5

For the detailed analysis of domain ontologies which constitutes the later stages of the DECOR Business-Knowledge Method, I committed to the respective parts of the IDEF family of methods for enterprise modelling and analysis.

IDEF (for Integrated Definition) is a set of modeling methods that can be used to describe operations in an enterprise. IDEF was created by the United States Air Force and is now being developed by Knowledge Based Systems Inc. Originally developed for the manufacturing area, IDEF methods have been adapted for wider use and for software development in general.

IDEF set of methods

Sixteen methods, from IDEF0 to IDEF14 (and including IDEF1X), are each designed to capture a particular type of information through modeling processes, e.g., IDEF0 for function modeling, IDEF1X for data modeling, IDEF14 for network design, etc.. IDEF methods are used to create graphical representations of various systems, analyze the model, create a model of a desired version of the system, and to aid in the transition from one to the other.

IDEF ontology design

IDEF5 is the method designed for Ontology Description Capture comprising the following steps which are equipped with more support and detail in the original documents:

- **Organizing and Scoping.** The organizing and scoping activity establishes the purpose, viewpoint, and context for the ontology development project, and assigns roles to the team members.
- **Data Collection.** During data collection, raw data needed for ontology development is acquired.
- **Data Analysis.** Data analysis involves analyzing the data to facilitate ontology extraction.
- **Initial Ontology Development.** The intitial ontology development activity develops a preliminary ontology from the data gathered.
- **Ontology Refinement and Validation.** The ontology is refined and

validated the ontology to complete the development process.

These steps, appropriately embedded and adapted, went into the respective steps of the DECOR method (see Section 4.3).

Besides the amalgamation of these two *de facto* standards and their embedding in the BPOKM environment, few changes have been made. The most important is that we did not adopt one of the there proposed process-modelling meta models, but introduced our own one. This was influenced to some extent by the ARIS methodology and set of tools.

ARIS

The worldwide successful ARIS methodology, as well as the ARIS-Toolset (an integrated business process reengineering tool), have been developed by the IDS Prof. Scheer Co., closely related to Saarbrücken University and its IWI Institute. ARIS offers step-by-step system development models via various business-process modeling and analysis activities. The ARIS methodology analyses a business process from three different perspectives, including the data view, the organization view, and the function view. This classification helps reducing the complexity of the process and allows a systematic and comprehensive analysis.

DECOR extensions

In DECOR, we took a similar perspective which motivated the different perspectives in our Business-Process Method. Further, the extended Event-Driven Process Chain as a process modelling paradigm and the look and feel of the ARIS Easy Design product showed us the direction in which we had to design our process modelling method, in order to stay comprehensible for the end users and usable for the consultants.

Used and potential input knowledge for DECOR

In Table 25 below, we list these three significant DECOR input streams together with the major reasons for using just this approach. Nevertheless, it should be noted that the major scientific and practical contribution is to amalgamate contributions from these three areas; not so much, to amalgamate *exactly* these approaches. At the time when this thesis was finished, there were also other, maybe equally suitable, candidates for inclusion in the DECOR method. Some of them have not been included because they were worse with respect to the reasons enumerated for CommonKADS, IDEF5 and ARIS, but most of them just were not yet so far developed when the work on the DECOR method was started. Some of those alternative candidates are listed also in Table 25.

	Used approach; background information	Reasons for incorporating this approach	Possible alternatives
Business Process Analysis Method	CommonKADS[96] [Schreiber et al., 1999]	• Comprehensive approach • Well documented and tested • Particular focus on knowledge	- ADONIS method - Memo [Frank, 2002] - ARIS method - ...
Business Process Modeling Approach	extended Event-Driven Process Chain (eEPC)	• widespread, well documented • easy to implement • known to project partners	- IDEF3 - ADONIS [Junginger et al., 2000] - INCOME/STAR [Oberweis et al., 1994] - Memo-ML [Frank, 1998] - UML [Oestereich et al., 2003] - MO²GO [Mertins et al., 2003a] - SOM [Ferstl & Sinz, 1993] - ...
Ontology Engineering	IDEF5[97]	• Long tested • Well documented • Contains quality measures for ontologies	- OTK method [Sure, 2003] - METHONTOLOGY [Gómez-Pérez, 1996] - KACTUS [Bernaras et al., 1996] - DOGMA [Spyns et al., 2002b ; Jarrar & Meersman, 2002] - Enterprise Ontology method [Uschold & King, 1995] - ...

Table 25: Methodological Input for DECOR

4.1.3 Overview of DECOR Solution Modules

To sum up this introductory section, we list the DECOR objectives "products", or solution modules, which will be described in more detail in the following sections:

- Objective 1: Enable sharing and reuse of process-related explicit knowledge.
  - ➢ DECOR Product 1:
    Process-oriented Knowledge Archive
- Objective 2: Ensure extensive, context-embedded exploitation and user-

---

[96] http://www.sics.se/ktm/projects/kads.html

[97] http://www.idef.com/

friendly access to Knowledge Archive content.

- DECOR Product 2:
  Workflow-triggered Knowledge Delivery Toolkit

- Objective 3: Support knowledge-oriented analysis of organisations and processes.

  - DECOR Product 3:
    Business Process & Knowledge Modelling Toolkit

Figure 4 gives a rough overview of the DECOR solution modules and their inter-relationships: The DECOR Business Knowledge Method is a business-process analysis, design, and modelling framework for integrated process and knowledge modelling. This methodological framework is supported by a modelling tool for graphically denoting the models of interest. The process models and domain ontologies created with these tools, instantiate the generic DECOR Basic Archive System to application-specific Process-Oriented Structured Archives. For these archives, (weakly-structured[98], if necessary) Workflow Models play the role of indexing structures organizing the knowledge contained. However, the workflow models can also be enacted by a Workflow Engine which, together with mechanisms for maintaining dynamic task context and task-specific archive access, forms the DECOR Workflow-Triggered Knowledge-Delivery Toolkit.

---

[98] A clarification: the concept of weakly-structured workflows was originally contained in the DECOR workplan. During project runtime it turned out that the cases did not require this feature. Hence we cancelled the topic for our tool and method development. Since there is definitely still much work to do, this is explained in detail in Section 5.3. Nevertheless, we kept the topic in this graphics shown in Figure 47, because: basically, a weakly-structured workflow is a set of activities with: (i) some activities can be further decomposed into sub-activities at runtime; (ii) the process logic is under-determined in the sense that not at each point in time of workflow execution, the process logic must necessarily be able to determine an unambiguous next activity: hence a human decision must be made; (iii) process-logic constraints might be deleted, added, or deleted at runtime, i.e. ad-hoc changes of the flow must be possible. Accepting this definition, it is clear that – though exhibiting an underspecified process logic and though being subject to change, normally, we can expect a process decomposition into tasks and subtasks, etc., at least to some level of depth. This can, of course, still serve as an organization model for structuring archive content. Even if only the task names would be available, they could already be used as semantic content indexes.

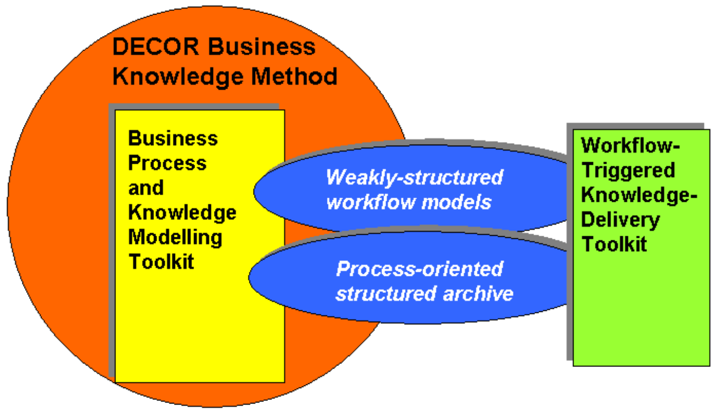


Figure 47: Top-Level View of DECOR Solution Modules

The DECOR approach for BPOKM presupposes a number things to be anaylsed and modelled: (1) Business process maps and domain ontologies for knowledge organisation and content description; (2) executable workflows for knowledge-intensive business processes; and (3) information flow (through context variables) and information needs for workflow enrichment. These models must be acquired and maintained over time. Further, the overall approach must be introduced in a company in the larger context of a comprehensive Knowledge Management or Business Process Management initiative. All required steps should be carried out by "normal consultants" in a "normal organisation", at reasonable costs, and with a predictable result. Hence, we need a structured approach for running Business-Process Oriented KM projects which supports all necessary project steps with appropriate methodological guidance and modelling tools. Figure 48 gives a slightly more detailed idea of the DECOR solution modules delivered to reach this goal.

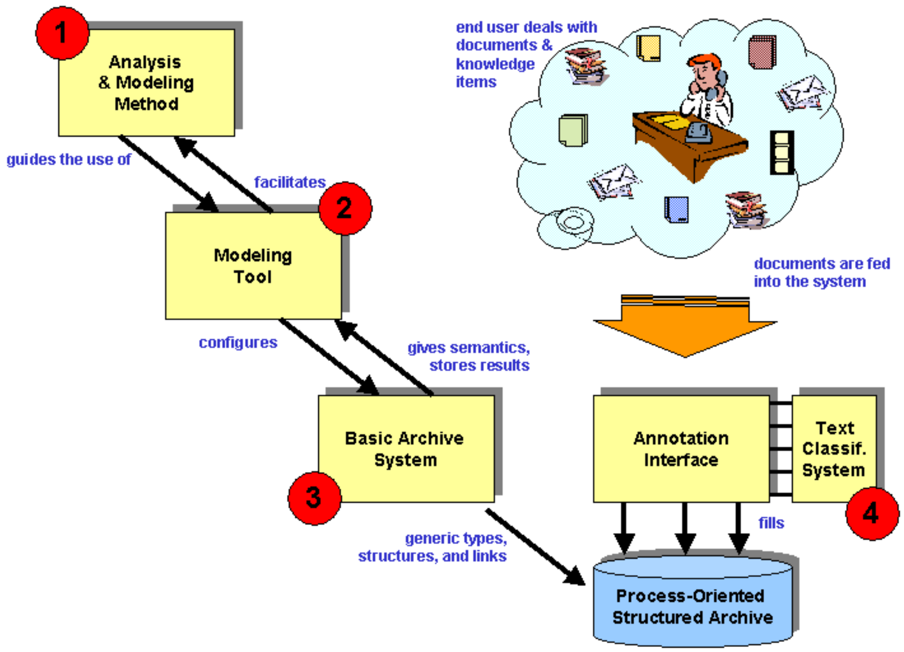


Figure 48: DECOR Modules Used at System Build Time

We briefly sketch these DECOR solution modules for system modelling and maintenance time (more details will follow in the next Subsections).

(1) DECOR Business Knowledge Method

DECOR's Business Knowledge Method provides a methodological approach for running BPOKM projects. Its main elements include:

A method for knowledge-oriented business-process analysis & design

- Identification of knowledge-intensive processes
- Process analysis and knowledge-oriented re-design
- Domain ontology construction
- Analysis of task-specific knowledge needs

The method combines elements from various sources, its procedural skeleton can be seen as an amalgamation of CommonKADS (Akkermans et al 1999; Schreiber et al 1999) and IDEF5.

(2) Business Knowledge Modeling Tool

The DECOR Modeling Tool supports in an integrated manner all modeling activities related to the method described above: business processes with task-specific information needs and process-specific context variables, as well as domain knowledge structures and ontologies. In contrast to most existing ontology

A process modelling tool for extended process models modeling tools, it primarily address users without a specific AI (Artificial Intelligence) background. It is oriented towards existing BPM tools (like ARIS™ or ADONIS™) and is realized as a set of related modeling methods for the commercial Microsoft VISIO® 2000 visualization tool. This ensures a wide usability of the software basis and a good familiarity of non-expert users with the overall look-and-feel. The VISIO® interface actions are coupled by a dynamic link to the DECOR Basic Archive System (see below). So, modeling activities at the user interface directly lead to the respective effects in the configuration of the underlying knowledge networks: new concepts or links are inserted in the ontologies, business process models are extended, or indexing concepts added to document models. This dynamic link with the Basic Archive System allows to equip the graphical modeling interface with a semantic foundation: For instance, only reasonable links are possible, i.e., links which do not respect the value restrictions of the represented relationship can directly be rejected.

(3) Basic Archive System

The DECOR Knowledge Description Layer: A content management system organized around process models

The Basic Archive System stores knowledge items plus metadata, as well as links between knowledge items. Metadata are represented in terms of underlying ontologies designed with DECOR modules (1) and (2). Business process models are one of many possible structuring criteria. Manual navigation in hierarchical indices extracted from index ontologies is allowed, as well as querying the archive by XML retrieval messages which combine retrieval constraints formulated over links and metadata. Software basis for the DECOR Basic Archive System is the CognoVision® product offered by DHC GmbH. CognoVision® allows to represent arbitrary knowledge networks built from attributed objects and attributed links, and to link information objects to structuring elements. Information objects encapsulate (i) logical content entitities like the set of all documents with the same content, but in a different language, and (ii) the related metadata.

(4) Annotation Interface

Semi-automatic indexing support

In order to fill the archive system, a software is needed for easily attaching semantic categories (in terms of modeled ontologies) to knowledge items. In this way, documents are fed into the process-oriented structured archive and indexed with metadata. Since indexing is a well-known bottleneck for ontology-based KM systems (indeed, for all document management systems, as well as for Semantic Web applications), in the run of the DECOR project a generic interface in the form of an API (Application Programming Interface) has been built between an annotation tool and an automatic text classification software, the MindAccess® tool provided by insiders information management GmbH. MindAccess® is an extensible multiple-paradigm tool which employs a number of state-of-the-art algorithms for text analysis and classification.

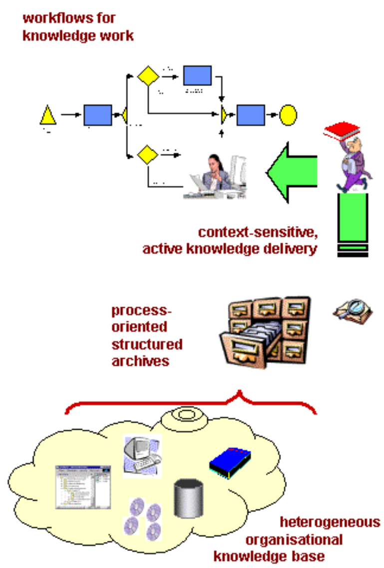


Figure 49: DECOR at Runtime

So far, we characterised the DECOR modules required for designing and installing a process-oriented structured archive and for filling it with annotated knowledge items. For the sake of completeness, I also mention in this overview the runtime modules required for process execution (although we know all these functionalities already from the KnowMore sample applications):

Modelling time must be complemented by enactment time

(5) DECOR workflow tool

The DECOR Workflow Engine (DECOR WFE) executes the process models specified at build time. In DECOR we laid special emphasis on (i) the deep integration with the DHC's commercial archive software, and (ii) on a comfortable system interface understandbale and usable by "normal end users".

The DECOR Application Layer

(6) Context-aware knowledge agents

The purpose of the DECOR Context-aware Knowledge Agents is to co-operate with workflow engine and modeled information needs, thus proactively offering information from the process-oriented structured archive to the user in charge of a

The DECOR Knowledge Broker Layer

certain task. It should be noted that the DECOR archive system offers not only possibilities for retrieving knowledge items at process enactment time, but also for creating documents and even folders, and for storing and indexed knowledge items from the running business process.

In the following, we will present the DECOR Process-Oriented Knowledge Archive as the basis of our technology platform. Then, the DECOR Modelling Method and Tool which is needed to configure and fill the archive. And finally, the DECOR Workflow Engine which shall exploit stored knowledge at process runtime.

4.2 The DECOR Process-Oriented Knowledge Archive[99]

The DECOR Process-Oriented Knowledge Archive was intended to manage manifold kinds of documents within an organization in such a way that they can be reused and exploited well, seen from a business process perspective. In the case that such a system would only act as a *meta-information system* that maintains links to information actually stored / contained in other systems, it would exactly realize the Knowledge Description Layer in our generic OMIS framework. Assumed that such a system could also store itself parts of the documents one is interested in, and that it will also offer some more or less sophisticated retrieval functionalities, it would also realize parts of the Knowledge Object Layer and of the Knowledge Broker Layer, respectively.

Requirements for Process-Oriented Knowledge Archive

Requirements.

When gathering requirements for such an archive system, within the realm of the DECOR project, we came up with the following general requirements for *powerful document data and metadata handling:*

a. **Meta data handling:** documents must be equipped with an extensible set of attributes to talk about them on the meta level (like trustworthiness, importance, actuality, or risks of some knowledge object – or its application).

b. **Multiple viewpoints:** document content categorization (indexing) must be possible wrt. multiple indexing dimensions; also, documents must be indexable with several concepts from the same dimension.

c. **Ontology-based indexing:** models for viewpoint characterization (e.g. indexing ontologies) must be allowed to be more complex structures than just lists or hierarchies of concepts.

[99] This Section is a slightly reworked and extended version of Chapter 2 and 4 of an unpublished manuscript written by the author of this thesis and colleagues from DHC GmbH [Herterich et al., 2003].

d. **Powerful link management.:** it should be possible to define specific relationships at the document level (follow-up document, new version, explanation, contradiction, …), at the ontology level (see under c: to define complex semantic networks), and between the levels (see under b: document refers to concept); also group links (1:n relationships) should be allowed.

e. **Flexible index handling:** it must be easy to change an indexing ontology (new concepts, structural changes) without far-reaching consequences for the whole document archive.

The CognoVision tool.

Simple publication and administration of information in inter-, intra- and extranet

CognoVision allows to manage global, enterprise-specific, or individual information, independent from system borders. It serves the purpose of systematic structuring of knowledge while avoiding the production of redundant information.

Creating and changing content

CognoVision is based on existing systems and uses established functionalities in order to administrate multi-media information created with different applications, and to create relations between these information objects. CognoVision aims at the realisation of a structured knowledge network, which assures the easy reuse of results for varying projects and tasks. The open architecture of the tool allows the simple and fast integration of *standard products* in the CognoVision workplace – such as MS Office products (Word, Excel, PowerPoint) and MS Visio. When working with CognoVision, the necessary documents and the associated editing functions are provided by the applications already used. CognoVision is installed above of these systems (like MS Office products, SAP, document management systems, file server, or the Internet) as a form of logical middleware, which correlates the source information in an intelligent and context-oriented. For doing so, CognoVision offers the modelling facilities and internal data structures which are shown in Figure 50 in a simplified manner.

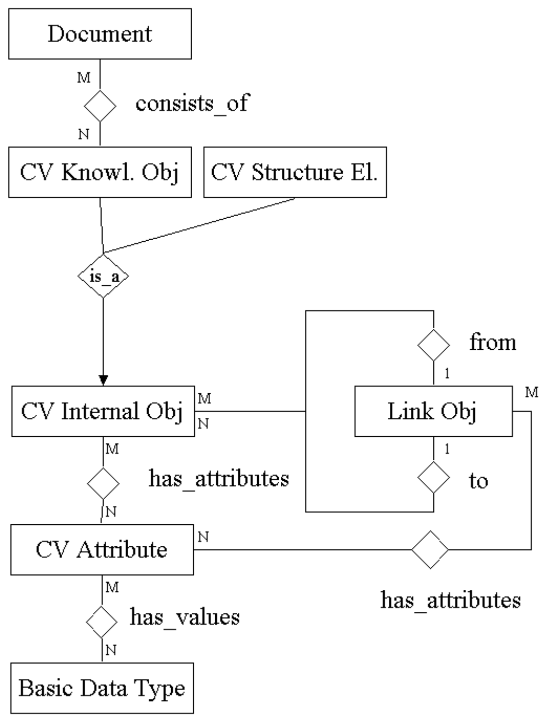


Figure 50: Metamodel of CognoVision Data Structures

- CognoVision basically distinguishes between Knowledge Objects and Structure Elements. Both are internally represented as CV Internal Objects.
- Structure Elements are the primary units for knowledge organization which can be embeeded into knowledge networks (see below) and which serve as structuring / indexing elements for Knowledge Objects (see below). Structure Elements can be sorted into a type hierarchy which is freely definable.

Structure elements correspond to ontological kinds / concepts / instances

  - Hence, Structure Elements may be interpreted semantically as Kinds / Concepts or Instances representing entities within Domain or Organization Ontologies.
- Knowledge Objects may cluster a number of electronic documents, such as office documents, operational data, or other other entities available in an organizational information system. They shall represent one logical information unit, clustered because it represents the same document in different languages, or different versions of the same document. Knowledge Objects can also be

Knowledge objects abstract from electronic data or documents sorted into a freely configurable set of Knowledge Object types, where a type is defined by its set of attributes which each instance may exhibit.

- Hence, CV Knowledge Objects can in a natural way stand for Knowledge Item Descriptions. (Side remark: CV Knowledge Objects can also be grouped in classes).

Attributes

Links can also have attributes

- All CV Internal Objects may possess Attributes with basic data types describing their domains. Attributes can be maintained for Knowledge Objects, Views, Structure Elements, as well as Links. Links can be classified. All attributes can be configured depending on the needs of enterprise using the DECOR tool. The following types of attributes are available:
  - *Name and type of object:* This attribute is necessary for the creation of an object.
  - *System attributes:* Are administrated automatically by CognoVision. Examples include: state (wrt. versioning model), original language, date of last change, time of last change, …
  - *Definable attributes:* When CognoVision is customized, one defines which attributes can be assigned to which type of object.
  - *Keywords:* Can be assigned to each object.

Search

- Attributes and keywords are the criteria for specifying search conditions in CognoVision. With the search results, a preview and all metadata (document properties) are displayed.
- Links establish a relationship between two CV Internal Objects, i.e.:

Links can associate two knowledge objects

  - *Between two CV Knowledge Objects*
    - This would correspond to a semantic relationship at the KDL level, e.g. for expressing that one document is required to understand the other; or that one presentation states the counterargument to the other.

Links attach semantic categories to knowledge objects

  - *Between a CV Structure Element and a CV Knowledge Object*
    - This is the typical semantic indexing function where a concept can be associated with a document through the KID.

- *Between two CV Structure Elements*
    - This opens the possibility to model arbitrary semantic networks for knowledge organization.

- The last important concept in CognoVision is that of a View.

View provide entry points and structures for navigation and browsing stored information

  - The user can define views using the corresponding Structure Elements (where the type hierarchy of Structure Elements induces a hierarchical tree-structure for browsing and navigation)
  - Documents are linked to Structure Elements and thus accessible via Views. Views may visualize application, project or user related aspects of information. Fields of knowledge which are correlated by their content, are reasonably condensed into one view.
  - Specific kinds of Views can be created automatically when one imports data from an other application. For example if one imports data from the ARIS toolset, groups, models and model objects generated in ARIS are imported to CognoVision as Views.

Figure 51 illustrates the basic concepts and relationships: CV information sources can be virtually anything available electronically in the organization. This set of information under the (partial) control of CognoVision constitues the Knowledge Object Layer of DECOR. The CognoVision information pool is the set of all Knowledge Networks created by Structured Elements, Knowledge Objects, Links and Attributes. This corresponds to the instance level of our general Knowledge Description Layer, i.e. the Knowledge Item Descriptions, as well as part of the underlying ontologies.[100] Finally, the CV views which are created for manual browsing and access, is designed for *human* interoperation, i.e., designed to be used at the Application Layer. But the view definition and manipulation itself is obviously another part of the Knowledge Description Layer, some interactive

---

[100]The missing part is mainly the Information Ontology which is defined in the CognoVision Administrator tool and specifies the data structures instantiated with these information pool elements. Moreover, of course, there is no formal ontology model behind the CognoVision approach. So, if one would like o have formal ontology reasoning as shown, e.g., in 3.2.2, one would have to attach an Ontology Management system as an add-on to this information pool layer.

methods may also belong to the Knowledge Broker Layer.

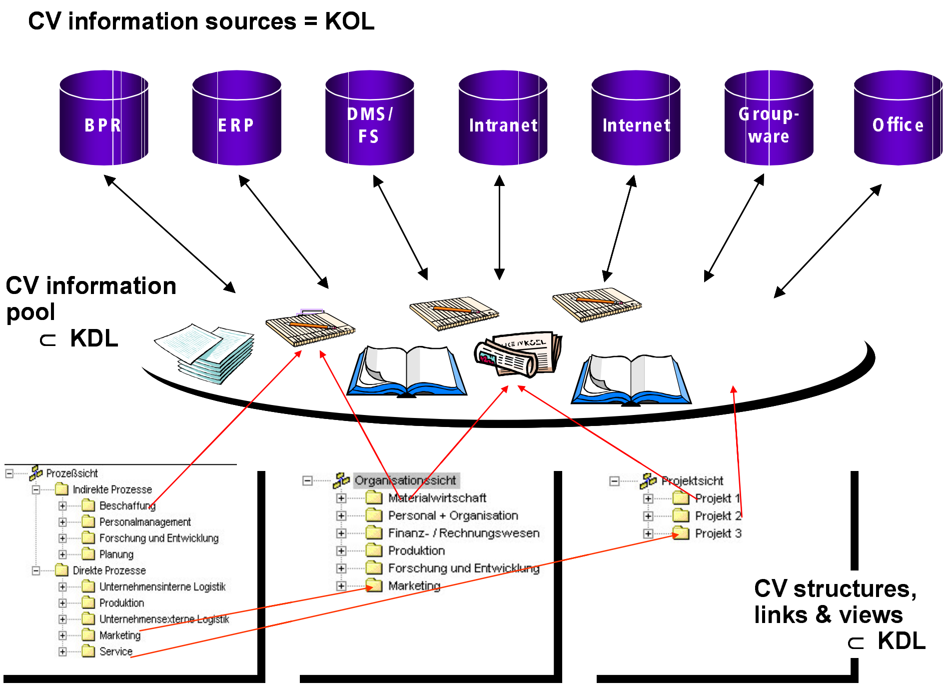


Figure 51: Knowledge Networks in CognoVision

Further functionalities covered by CognoVision are: version management and multilinguality for all objects in the system; multi user management; an access control management (rights of individual users wrt. to specific objects / object types).

Coupling CognoVision to mindAccess.

Metadata creation may become a tremendous bottleneck

Our experience in many research and industry transfer projects shows clearly that potential customers of knowledge and information management approaches which are based on explicit metadata, are largely afraid of costs and effort for metadata creation (i.e. indexing), both (i) during system use and maintenance, and (ii) during system set-up, for importing existing, typically huge, document and information corpora. Hence, as indicated in Figure 48, we experimented in DECOR with automated text classification approaches for metadata creation.

To this end, we implemented[101] a generic interface system between document management and text classification systems, the IDA (Intelligent Document Access) tool. IDA was aimed at providing a powerful interface between arbitrary document management and arbitrary text categorization tools. To this end, it offers a tool-independent text categorization API (Application Programming Interface) accessible from tools like CognoVision. We prototypically linked the mind-Access® SDK into our system, by mapping the generic API functions onto mindAccess operations. mindAccess - provided by insiders information management GmbH - is an extensible, multi-paradigm tool which employs a number of state-of-the-art algorithms for text-document retrieval, comparison, categorization, clustering, etc. Since the topic of document indexing is not the center of this thesis, we won't go in any detail about IDA. Some more information can be found in [Decor, 2002].

The IDA tool is a generic interface between document management and automatic text categorization

Now that we explained the CognoVision system which represents the main technical backbone of the DECOR Process-Oriented Archive System, let us come to the methodological aspects. In the next Subsection, we will see the DECOR method for organizational analysis, take-up and planning, which is equipped with a modelling tool, which in turn can be understood as a customization tool for the Process-Oriented Knowledge Archive.

Next, we explain the analysis and modelling method, before the modelling tool will lead us back to CognoVision.

[101] This work was done relatively independent by Tino Sarodnik, who implemented an excellent piece of software for this purpose.

4.3 The DECOR Business Knowledge Method and Tool

4.3.1 The DECOR Business Knowledge Method

The method shall guide BPOKM projects

There should be a structured approach for performing business process-oriented knowledge management projects which supports all necessary phases with appropriate methodological guidance and tools. The DECOR Business Knowledge Method targets at that objective by amalgamating elements from the CommonKADS [Akkermans et al.(1999), Schreiber et al. (1999)] and the IDEF5 [IDEF5 (2000)] methods. Figure 52 provides an overview of the method's steps.

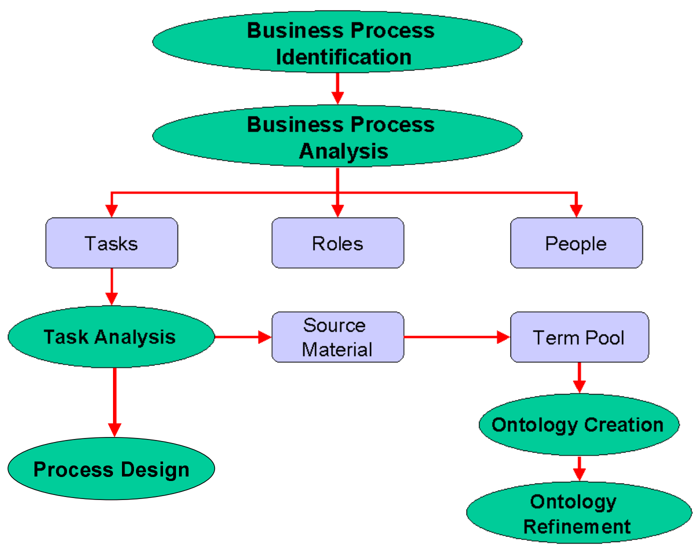


Figure 52: Overview of the DECOR Business Knowledge Method

Overview of DECOR Business Knowledge Method

Before we go through these steps with some more detail, let us give a brief overview: After (1) identification of the appropriate process to be supported by a KM project, this process is (2) analysed in detail. Detailed process analysis follows CommonKADS and elucidates – like most process analysis methods – involved tasks, roles, and people. Then, the tasks are focussed at in the (3) task analysis step. Typically, one asks for inputs, outputs, performer, pre- and postconditions, etc. We lay particular emphasis on the knowledge perspective which analyses knowledge needs and knowledge contributions produced by a certain task – this is not unusual when designing a Knowledge-Based System, but normally rarely done in conventional BPM/BPR projects. In the run of this task analysis, a number of documents are identified and produced, respectively, which are then input for the (5) ontology creation and (6) ontology refinement steps which are borrowed from the IDEF5 approach. This approach starts from terms, collected from source material, in order to identify concepts, relationships, and instances for defining the ontology. In parallel to ontology creation, and partially intertwined with task analysis, process step (4) takes place, the knowledge-oriented process re-design which aims at an optimized process layout, seen from the Knowledge Management perspective.

Now, we discuss these steps exhaustively.

Business Knowledge Method, Step 1

Step 1: Business Process Identification.

In order to identify the most appropriate business process(es) to be focussed on in a BPOKM project, the following questions should be considered:

How can an "interesting" target business process be identified?

1. Is the process mission-critical and does it require improvement?
    - First of all, this is not a DECOR-related question. To decide this question, standard BPM/BPR methods can be employed. Often, end users really know already where their core problems lie. In the case that a process is really important and problematic, and thus it requires improvement, we can ask the next question:
2. Is the process knowledge-intensive?
    - To identify whether a process is indeed a knowledge-intensive process, some relatively rough, but easy to assess criteria may be applied, such as knowledge intensity, process complexity, wicked-problem criteria (see below for more details). Of course, if it is not knowledge-intensive, it might nevertheless be a good target for process reengineering. But then we are out of the scope of DECOR. In the case, that we have identified a knowledge-intensive process, we proceed with question 3.
3. Is there knowledge-related improvement potential?

- First we have to accomplish an *as-is* analysis of the current business process which is extended by KM-related topics and criteria. Then we can go for a *to-be* design. Besides standard redesign measures, this should include a knowledge-oriented redesign as discussed below. In principle, these two steps (analysis and redesign) refer already to the next steps in our methodological approach. However, in this phase here, we need at least already some rough idea about current and future status for assessing expected benefits and effort – which is question 4.

4. Is a BPOKM project economically promising?
   - Since this thesis focusses on technological and not on economic issues, we did not go deeper into this issue. However, one could apply, for instance, the procedure proposed by the CommonKADS method for realizing a feasibility study [Akkermans et al., 1999b].

Hence, the two middle questions, 2. and 3. are to be clarified a bit more in detail.

Criteria for the identification of knowledge-intensive business processes.

How to assess the knowledge-intensity of a business process?

According to [Remus, 2002], we can compile a catalogue of criteria for the identification of knowledge-intensive business processes, in analogy to the criteria catalogues proposed by [Goesmann et al., 1998] or [Becker et al., 1999] for assessing the workflow-support potential of business processes. Such a list of criteria should not be understood as an instrument for a strict distinction between knowledge-intensive and "ordinary" processes, it is rather an indicator that some further analysis might make sense in a certain area. We translated and slightly adapted Remus' approach as shown in Table 26 below and extended it by elements found in [Müller et al., 2004].

Business Process Perspective	Process criteria	Typical properties of knowledge-intensive processes
Process environment	Organisation + culture	○ Knowledge sharing culture ○ Knowledge-oriented incentive systems ○ KM roles and processes

	Organisational environment	○ Knowledge-intensive industry & competitors (high-tech company, ...)
	Cross-process interdependencies	○ Complex interdependencies with other processes
General process characteristics	Process complexity	○ High complexity (many conditional branches, parallel threads, loops, ...)
	Variability	○ Many special cases (control flow not exactly predictable)
	Structuredness	○ Weakly structured
	Participants	○ Many participants ○ Interdisciplinary ○ experts
	Process objects	○ knowledge-intensive products and services
	Controlling	○ vague objectives, unclear how to measure
	Process instantiations / business cases	○ seldom instantiated, long-running
	Process type	○ individual cases, issue-related cases, ad hoc[102] ○ „typical" knowledge-intensive business processes (like innovation, R&D, product development, management processes, improvement processes)[103]
Task characteristics	Controlling	○ vague objectives and measurement
	Training time	○ relatively long
	Workplace setup	○ typically looks chaotic
	Task / acticivity type	○ communication and information oriented, argumentation based ○ individual cases, issue-related case ○ typical tasks: decision-making, problem-solving, analysis & assessment, management
	Collaboration aspects	○ important and discussion-oriented, large and changing groups, distributed work groups
Employee specific characteristics	Decision-related scope of discretion	○ high degree of autonomy in work organization and decisions ○ significant influence of employee on process results
	Rules and regulations	○ unstructured, individual working rules

[102] Terminology (though translated) follows the classification by [Picot & Rohrbach, 1995].

[103] For a selection of "typical" knowledge-intensive business processes, see [Davenport et al., 1996] and [Eppler et al., 1999].

	Skills + competences	○ learning, creativity, innovation[104]
Resource-oriented characteristics	KM instruments and systems	○ KM instruments and tools are used
	Knowledge representation	○ Manifold forms used: DB entries, documents, hypertext, multimedia, drawings, ...
	Knowledge exchange	○ Often informal, face-to-face, document-based (presentations, memos, mails), ...
	Knowledge types	○ Process knowledge about, from and within the process
	Knowledge access	○ Typically difficult (technical knowledge, individual judgments, tacit knowledge)
	Used information sources	○ Manifold, high volume, frequently changing, different formats
	Complexity	○ High, context-dependent, case specific
	Actuality / time	○ Short knowledge half-life time, need for continuous "knowledge maintenance"
	Budget	○ Typically high

Table 26: Characteristics of Knowledge-Intensive Processes

How to assess the knowledge-related improvement potential of a business process?

Detection of improvement potential.

Having identified a critical process which requires improvements and is knowledge-intensive, the question arises whether KM issues are really the critical matter. Though this should be relatively easy to see in reality, and a general procedure will never replace individual consulting experience and talent, we give nevertheless a list of indicators which could provide hints to KM-related improvement potential (Table 27). The list should not be considered exhaustive, nor even well-organized or free of redundancy. It is rather an internal checklist to create awareness for potentially occuring phenomena which should be focussed on when arising. The list of indicators is compiled from inputs found in [Allweyer, 1998] and [Müller et al., 2004] and coming from Eppler.[105]

[104] cp. [Drucker, 1999]

[105] Martin Eppler, Hochschule St. Gallen / Switzerland (personal communication).

• Preparation of standards, evaluations, projects, proposals, etc. which do already exist in the organization
• Strategically important knowledge areas are not covered by the organization / organizational processes; unsatisfied knowledge needs
• No knowledge sharing culture visible
• Knowledge monopolies, i.e. important knowledge which is owned by only one / few employee(s)
• Existence of not used or underexploited organizational knowledge
• Building up skills and know-how which is already available in the organization; multiple creation / acquisition of the same knowledge
• Creation or acquisition of knowledge which is not required or not used
• Employees' knowledge profiles are insufficient
• Buying licences and services though there are own developments
• Information overload at all levels
• Internal experts are not identified
• Use of old or inappropriate knowledge
• Missing integrated IT infrastructure for knowledge logistics
• Preparation of knowledge not appropriate for the users addressed
• Expensive searches for information, complicated knowledge access
• Missing links between operational information systems (such as production databases, workflow, document management, CAD, ...) and dedicated KM systems (such as skill management, lessons learned, innovation management tools, ...)
• Project experiences are not systematically documented and reused
• Mission-critical knowledge is lost by personnel fluctuation
• Insufficient transparency wrt. external, relevant knowledge (documents, experts, trends, developments, patents, ...)
• Barriers for knowledge sharing such as lack of time or missing incentives
• Organizational weak spots which hinder knowledge sharing and optimum knowledge reuse

Table 27: Indicators for KM Problems

Now having decided to focus on a certain process, the next step of our method is undertaken.

Business Knowledge Method, Step 2

Step 2: Business Process Analysis

This step involves a general description of the selected business process(es) in terms of (a) tasks performed in the business process; (b) roles involved; (c) key people and (d) source material[106]. This step is necessary in order to establish a comprehensive description of the specific business process, to prepare process (re-) design and detailed task analysis.

In order to support this overall process analysis, we adopted mostly the CommonKADS questionnaires and forms for the respective analysis activities and changed them slightly in order to prepare the DECOR-specific subsequent steps and in order to be consistent in terminology. The reworked questionnaires can be found in the DECOR project deliverables. They were mostly prepared by the DECOR team at the ICCS institute of the National Technical University of Athens (NTUA).

Step 3: Task Analysis

Business Knowledge Method, Step 3

This activity is concerned with a more detailed description of the individual tasks, including (i) their input and output objects, (ii) the source material handled within or delivered by the task, (iii) control relations between tasks along with constraints that govern the execution of each task, as well as (iv) the roles performing the task, and so on. Moreover, every task in the process is assessed through its contribution to the core activities of Knowledge Management, i.e. generate, store, distribute and apply knowledge.

Again, we adopted the respective CommonKADS questionaire as documented in the Appendix. The major extension is to ask for material which can lead to source material for ontology creation. Some hints for performing such interviews:

Have the people in the boat – really!

- As a side remark: before start, make clear what the overall goal of the procedure is. In the ideal case, the task analysis phase is part of an officially announced and promoted project that follows the guidelines of reasonable change management projects. For KM projects, such guidelines have been provided by many authors, e.g. [Mentzas et al, 2002; Wiig, 1995], and include topics such as starting with a Kick-Off workshop,

---

[106] Source material is a technical term for the ontology modelling method meaning textual sources which enable the identification of relevant terms and concepts in the problem domain area that is tackled. It will primarily be identified in the task analysis step, but some key materials may lso be identified at the level of overall process analysis.

having both the floor people and the top management in the boat, let the involved people share your mission, vision, values, and goals, ... Practically, this comprises also simple things like: Before you try to design an ontology with some employee never heard about the topic, show him a live demo of an easy to understand BPOKM software demonstrator, explain the functioning of the approach, and explain the underlying ontological foundations.

Work case-oriented, example-driven

- Ask for knowledge domains and their specific instances which are relevant for performing the task affected. This can be done best when going through concrete case examples.

Find all text sources somehow related to the task

- To do so, ask for relevant input or output material in the form of texts, documents, etc. which shape the knowledge area affected. Ask for background knowledge sources in the form of company regulations and handbooks, legal or regulatory texts, typical case examples, operational documents and supporting material, consider schemata of databases involved and forms affected, examine structuring criteria of Intranet areas, department structures / job descriptions or document management and archive systems which are affected by or knowledgeable in the area of the task under consideration.

Brainstorm about structures when gathering material

- Try to brainstorm about conceptual structures organizing the field of the application domain. Challenge "modelling decisions" always with "why" questions, ask for sub- and superconcepts, as well as similar topics. The results of such brainstorming discussions constitute an own, important class of source material for ontology creation.

Get a multi-perspective view, involve different people

- Focus the interview not only to the actual case mover doing the task every day, but ask also his boss or colleagues, employees who perform the task seldom, in the ideal case, ask also affected roles (e.g. the "customer" of a task, the owners of precedent of subsequent tasks). Ask for the best introductory material that somebody could get who is new in the job.

Ontologies are more than a data schema

- Don't stop with a data schema of the data involved. The knowledge structures involve far more. If you get stuck and the problem area requires really deep domain and problem-solving knowledge, think about traditional methods and tools for Knowledge Acquisition in Expert Systems (from methodological advice – thinking aloud, structured interviews, ... – up to technical support with tools like repertory grids, etc, [Speel et al., 1999]).

- Ask for the knowledge that is created or searched for in such a task. In a routine instantiation, and in "special cases". What was the most difficult case which came, up to now? Why was it difficult? To whom is a too difficult case forwarded? How would be asked for help? Why? Is created knowledge stored? Where, how?

Elucidate knowledge needs and contributions

- Are there personal notes / remarks / commentaries to the official regulations? What are they talking about? What might distinguish a very experienced and successful case mover from a newcomer?

Don't oversee the "inofficial" side

Essentially, the knowledge-oriented extensions are all centered around knowledge needed and knowledge produced. The output of this step should be an optimum input for (i) domain ontology creation; (ii) information ontology creation; (iii) extended task modeling for introducing support requests, KM tasks, etc. Hence we can understand the goal as a generalized information need analysis (generalized insofar as one would have to tackle these questions both for required and for produced knowledge) as it is known from Information Engineering (see Figure 53).

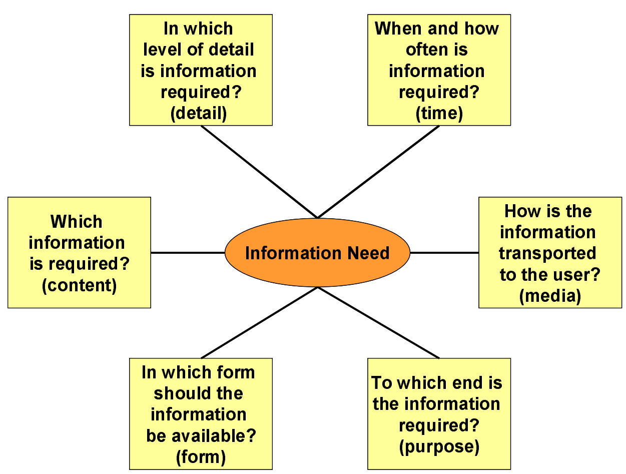


Figure 53: General Domain of Information Need Analysis[107]

As a side remark, it should be noted that – whilst a separate step in theory – in

[107] Figure follows [Michelson, 2001]. Translated from [Weih, 2002].

practice, this collecting of source material is already the first part of the Ontology Engineering. Hence, all insights produced in that area (cp. [Sure, 2003]) might also be useful here.

Business Knowledge Method, Step 4

Knowledge-oriented reengineering is reengineering ...

Step 4: Business Process Design

First, we have to note that knowledge-oriented (re-) design is, of course, in particular, "normal" (re-) design, i.e. when designing a process optimized with respect to KM, we should also include the traditional reengineering optimizations. Hence, well-known reengineering principles like those formulated by Hammer should also be regarded: "capture information once, and at the source; put the decision point where the work is performed; link parallel activities instead of integrating their results; ...." etc.[108]. However, since Business-Process Reengineering is not the topic of this thesis, we won't go into details here.

... and is highly interwoven with task analysis and ontology engineering

Regarding the position on the overall DECOR Business Knowledge Method we can say that this step is closely related to the results of task analysis, can be done partially in parallel and intertwined with Ontology Creation and Refinement, but at the end, needs the final Domain and Information Ontologies for clearly stating Support Requests with Information Needs, referring to the both. The Business Process Design includes modelling the business process with the DECOR Modelling Tool to be explained below. The output of this step is an executable business process model enhanced with Knowledge Management tasks for improving knowledge flow, sharing and reuse in the business process. The activity can roughly be divided in three sub-activities:

(4a) Refined task analysis

(4b) Planning improvements

(4c) Implementation of improvements.

(4a) Refined task analysis.

The refined task analysis considers primarily the knowledge-intensive tasks of a

---

[108] [Hammer, 1990], cited from [O'Leary & Selfridge, 2000].

business process. It settles upon the results of the task analysis already done and aims at an in-depth understanding of knowledge to be used and knowledge to be produced in this specific task. In order to come to such an improved understanding, we can follow Heisig & colleagues' (Fraunhofer IPK Berlin) approach, to analyse all interesting tasks from two perspectives [Mertins et al., 2003b]:

Refined task analysis follows the IPK approach

- First, take the **Knowledge Demand Perspective**.This comprises to analyse which knowledge is required, at which quality, where it may come from, and which prior KM activities must be ensured to guarantee a closed, functioning knowledge supply chain.
- Then, take the **Knowledge Supply Perspective**. Here, it is analysed which knowledge is produced in a task and whether it finds its way to potential users, i.e. whether all KM roles and activities are in place such that the supplied knowledge may somewhere meet a demand.[109]

Depending on the question how difficult and how important exact and comprehensive results of this step are[110], more sophisticated instruments and analysis methods are possible to increase the level of quality achieved in this step, for instance:

  - A **communication-structure analysis** of the organization which helps to disclosure hidden networks of collaboration, communication and knowledge sharing which might not even be consciously noticed (cp. [Dämmig et al., 2002]).
  - Advanced knowledge-acquisition methods can be employed for identifying related domain ontology concepts etc. (cp. [Speel et al., 1999])

- A third perspective that could be taken for finding optimization needs and opportunities has been heavily investigated in the Promote

[109] We may remember that this overall approach is similar to the generalized information need analysis from Information Engineering that I mentioned already earlier [Michelson, 2001]. It refers at least to content (Domain Ontology) and media (Information Ontology) aspects.

[110] This varies a bit with the importance, criticality and complexity of the process investigated; and with the question whether the first implementation shall already cover all possible support aspects, or, contrarily, shall start with "quick wins" to be extended in an evolutionary improvement process later.

project [Woitsch & Karagiannis, 2003; Hinkelmann et al., 2002], could be called the **Process-link Perspective**. There, the focus is on identifying improvement potential by systematically check possible synergies with other processes, with a stepwisely increased scope of view:

- **Instance Scope:** are there other activities within this process that are not yet linked through a KM task or an information flow, and which could require/offer knowledge which is created/needed here?
- **Intra-Process Scope:** is it possible to reuse experience or provide experience for reuse between this actual process instance and a former or a future instantiation? This means, should we query or fill, e.g., a process- or prcess-step specific lessons learned database?
- **Inter-Process Scope:** is in the process currently under examination, some knowledge created or required which could be consumed or produced within another business process in the organization (another process model, not another process instance!).
- **External Sources:** finally, one comes to the outest look, to ask whether information outside the organization can/must be linked into the process.

In each cases, if there are possible connections, which should be exploited, it is an option to create KM tasks for establishing the respective links. This Promote view adds very explicitly the facet of creating knowledge-sharing process networks as a structure orthogonal to the existing business processes.

Static-dynamic + local+global as an additional analysis dimension

A last facet which can be examined for improving the refined task analysis is the systematization of information and knowledge needs as introduced in Subsection 3.1.2. In order not to forget or oversee interesting ideas, it might also make sense to extend the analysis by a systematic stepping through the basic determinants and possible influence factors of information needs: What are the static, local information needs, what are the static, global informations, and so on?

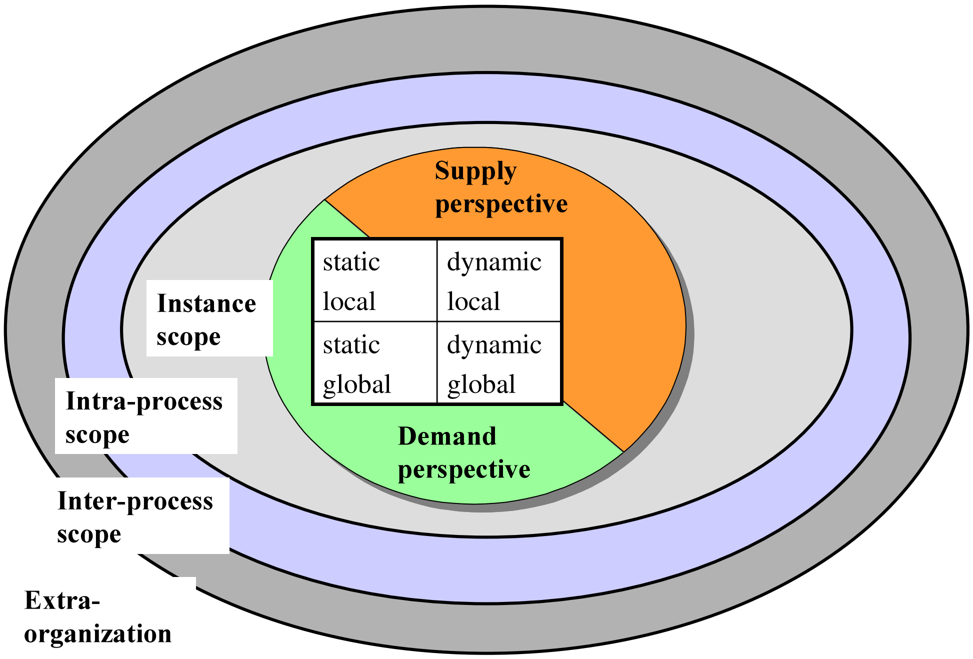


Figure 54: Multi-Perspective Refined Task Analysis

Figure 54 combines the several perspectives of refined task analysis in a coherent analysis framework.

(4b) Planning of process improvements.

Based on the assessment of the knowledge demands and knowledge supplies, as well as potential partners for establishing links, the realization of these improvement steps must be considered in more detail. At least two issues must be tackled:

a) **Identification of barriers and hurdles:** As [Oberweis et al., 2001] point out, there may exist hurdles and barriers for successful KM implementation which are orthogonal to the process aspects already discussed. Examples could be inappropriate organizational structure, wrong workplace design, cultural issues, hindering departmental structures and responsibilities, etc.

b) **Selection of appropriate instruments:** Besides the direct implementation means discussed below, some of the identified improvement potentials could be realized using standardized, more coarse-grained approaches. In particular, the following is possible:

- Select from a palette of intra-organizationally acknowledged and/or generally accepted **best practices** or **good practices** (for instance, the IPK approach to KM is pretty successful in commercial projects by integrating into business processes instruments which are – as best practice experience – known to yield good results for this specific business process [Heisig, 2001; Mertins et al. 2003]).

  Related / identical is Heisig's suggestion to link the to-be-improved business process with exsting KM Processes, such as Skill Management, Continuous Process Improvement, etc.

- Another – somehow similar – approach is to select, adapt and install **KM reference processes.** Though KM reference processes are by far not yet existing or even generally accepted, there exist neverthe-less promising approaches, such as presented by [Polterauer & Mayrhofer, 1999; Blessing et al., 2001]

(4c) Implementation of improvements.

For realizing the so-identified and detailed process improvements, a number of concrete steps can be undertaken:

- **Add KM tasks:** Some knowledge-related tasks to be included, in particular information retrieval and gathering tasks, can be fully or partially automated by accessing Mnemonic OMIS functions by Support Requests (this is the "typical" KnowMore solution). In some cases it might be appropriate to add a new Mnemonic Process to the OMIS KBL, if some complex OMIS-manipulating activity is often called repeatedly in several processes.
- **Close knowledge cycles:** If a gap in the sequence of knowledge-related tasks is identified, it is filled by adding the corresponding tasks. For example, if somewhere in the process the generation of knowledge has been identified but this knowledge is not stored, a KM or an ordinary task for storing this knowledge can be added in the business process (this goes essentially back to Heisig's GPO-WM® method [Heisig, 2001]).

- **Add KM sub-processes:** Some operative BPs can easily be enhanced by interleaving them with (standardized) KM processes or process parts. (this is mostly the Promote approach [Karagiannis & Telesko, 2000]).

Last, but not least, it must be noted that in this thesis, we naturally focussed on technological issues. Of course, all these steps must be complemented and supported by appropriate means of organization structures, roles, responsibilities, etc. (cp. [Allweyer, 1998]).

Business Knowledge Method, Step 5

Step 5: Ontology Creation

This activity involves the development of a preliminary ontology, taking as input the source material – and maybe preliminary term collection – gathered during the preceding steps. It follows completely the well-documented IDEF5 procedure, such that we don't have to discuss or explain it in much detail..

Data analysis with simple paper-based forms

It consists of a data collection process which is both iterative and interactive (between process stakeholders and consultant / ontology engineer) process. The data collection may occur in different modes (interviews with domain experts, direct transcription of data from source documents etc). It might be supported by text analysis methods [Maedche, 2002] or integrated with brainstorming approaches [Sure, 2003]. In the DECOR case studies, we followed a simple, fully manual process.

Data analysis as a stepwise, traceable, identification of relevant entities in the domain

Regardless of the data collection methods used, each piece of collected data must be traceable back to its source, because it is this data that provides objective evidence for the basic ontology structures that are later isolated from this data. Therefore we use employed types of documents to facilitate source data traceability, as it is suggested in the IDEF method (and documented in the Appendex in Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**)[111]: 1) Source Material Index, 2) Source Material Description Form, 3) Term Pool, and 4) Term Description Form.

In the Term Pool and Term description form, we record the meaningful Terms

[111] Please note: as we mentioned already earler, our approach will principally work with most known Ontology Engineering methods, The relevant innovation is the set the focus of the difficult analysis task by the process analysis' task analysis, and to feed task analysis documents into the ontology creation pipeline.

relevant for ontology development. It is from these Terms that we construct an initial ("first pass") characterization of the ontology, i.e. identify candidates for the central entities comprising an ontology:

- **Kinds / concepts / classes:** can be considered as an objective category of objects sharing a set of properties;
- **Instances / objects:** concrete entity in the real-world; belongs to one or more specific kinds
- **Characteristics / attributes:** are the properties belonging to a Kind;
- **Relations:** are the sorts of general features that Kinds / Instances exhibit jointly rather than individually – with a particular importance of the taxonomic relationships which describe subclass-superclass relations between Kinds; and of the instantiation relationships which relate instances and kinds.

Step 6: Ontology Refinement.

Business Knowledge Method, Step 6

This activity involves the refinement and validation of the ontology. While the Ontology Creation step is merely concerned with abstract structures, i.e., Kinds, Attributes, and Relations, during this step, the ontology structures are instantiated (which means also: tested) with actual data. The result of the instantiation is compared with the ontology structure. If the comparison produces any mismatch, every such mismatch must be adequately resolved. Refinements (if any) to the initial ontology are incorporated to obtain a validated ontology.

After all, wee see that Step 4 (Business Process Design) represents the central working step for improving processes to include KM activities. Within this step, all modelling activities are based upon the process modelling formalism and tool which are described in the following subsection.

4.3.2 The DECOR Modelling Tool[112]

The DECOR Modelling tool is built upon two software systems:

---

[112]The development and presentation of the DECOR Modelling Tool was done in collaboration with colleagues from DHC GmbH Saarbrücken (mainly preparing the VISIO visualization) and NTUA Athens (preparing the modelling examples using the IKA case)

- the DHC CognoVision® tool [Müller & Herterich, 2001] for document and metadata handling, and
- Microsoft's VISIO® tool for visual modelling of the appropriate CognoVision structures.

Coupling CognoVision and VISIO

The two software systems are integrated as follows: Via a dynamic link, MS VISIO maps the VISIO modelling constructs (i.e. the several shapes and arrows we use for drawing pictures) to entities (Structure Elements, Knowledge Objects, or Links) in CognoVision. Thus, shapes in VISIO become CV Knowledge Objects, and edges become CV Links. Method-based modelling means that only a restricted set of shapes is allowed, and only a defined set of links is allowed for connecting a given pair of shape types. Since shapes and edges in the VISIO model are – during the process of modelling – mapped to objects and links in the CognoVision database, CognoVision can check that the link types are consistent with the selected modelling method.

Enactment of workflow models

The so-developed process / workflow models can later be enacted using the DECOR workflow engine. For that, all required information must be modelled in VISIO such that it can be stored as attributes in CognoVision.

Available modelling primitives

DECOR workflow metamodel.

Now coming to the functionalities of the DECOR Modelling Tool: In order to model knowledge-related tasks and knowledge objects within knowledge-intensive business processes, we construct a process metamodel that is closely related to the set of concepts and definitions in Chapter 0, but extended in such a way that it can directly be enacted as a workflow – i.e. we have to introduce, e.g., explicit modelling primitives for control and data flow. Further, the model shall emphasize the relationships between between the process and the knowledge perspectives (cp. [Papavassiliou et al., 2002]).

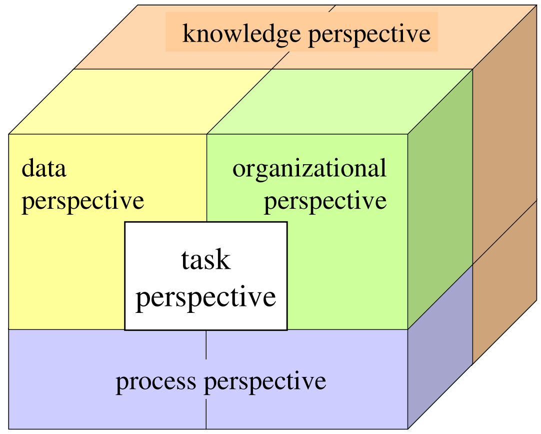


Figure 55: Workflow Modeling Perspectives

The knowledge perspective is a new, orthogonal layer

The Figure 55 – which extends an illustration from [Aalst, 2002] can be understood as follows: Van der Aalst's original illustration is the forepart – tasks stand at the center of our considerations; they have links to organizational issues / resources and to data / information objects; the process perspective models the business logic behind the process which combines all three elements into a purposeful whole. Now, another orthogonal layer is introduced which is linked to all other perspectives: the knowledge perspective models in its domain ontology parts of the functional knowledge dealt with in activities, it may give semantics to organizational and to data models, and it is linked through input/output relationships and via context and decision variables to the process logic.

The basic modelling constructs that are provided are shown in Figure 56 and are described as follows:

- **Tasks:** A task represents the structured work in the business process that must be done to achieve some objectives (it corresponds to an ACTIVITY in our definitional framework). We can distinguish:
    - *Operative Tasks:* represent MANUAL ACTIVITIES in our ontology
    - *Knowledge Management Tasks:* KM Tasks are used to describe the work associated with the generation and application of knowledge in the business process. The execution of a KM task may contribute to the successful performance of an operative task. They correspond to KM TASKs in the ontology.

- *Interface Tasks:* An Interface Task is a special kind of task just for modelling convenience. It is used to connect two different models by linking to the start of a more complex model seen here as a black box.
- *Automatic Tasks:* An automatic task describes work that can be done without any user interaction. It is an AUTOMATIC TASK in the ontology.

- **Events:** Events are used to trigger the execution of tasks.
- **Connectors:** Are used for modelling the logic of the business process.
- **Data Objects**: They describe variables used in the model to control the flow of the business process when executed by the workflow engine.
- **Knowledge Objects:** Knowledge Objects represent the explicit knowledge required in a specific business process. They serve as input or output for Tasks and KM Tasks.
- **Roles**: Tasks and KM Tasks are assigned to roles during the modelling of the business process. They represent ORGANIZATIONAL ROLEs.
- **Persons:** Persons describe real employees, the users of the tool. If the model is enacted, persons are playing the roles that have been modelled.

Task and organizational modelling perspectives

A knowledge-intensive business process is defined in a workflow model. The workflow model consists of tasks and their interdependencies. Each of these tasks can be decomposed into (sub)tasks, which in turn can represent a whole workflow. The proposed task and organizational perspectives are depicted in Figure 56.

Tasks of the workflow model are assigned to roles during modelling. Each of these roles has a set of permissions associated regarding the usage of the organisation's resources (tools, applications, etc.).

Control perspective

To cope with the **control perspective** of workflow modelling, we make the following provisions: Tasks are connected with events using control flow elements (sequence, and, or, xor) forming Event-driven Process chains (EPCs). EPCs are extended by links to other relevant entities. In this way, tasks can be connected to input and output data to model the data flow in the process and to knowledge objects to model the information flow. The control flow of the business process is modelled using sequences, splitters and joiners. With the sequence flow element, it is possible to link two tasks sequentially. More interesting are the split-join constructions that allow a path in the process to split into multiple parallel branches. It can be specified that such parallel branches all have to be executed at the same time (and-split), that only one (xor-split) or some (or-split) of these branches have to be executed.

Knowledge perspective

In order to support in an integrated manner the modelling of those activities that are associated with the creation and application of knowledge, we extend the EPCs with additional tasks, the Knowledge Management tasks. The usage of these tasks has already been explained above. These KM tasks, together with their control flow and the context variables to control their specific behaviour, are one major part of the **knowledge perspective**. The other important part of this perspective are Ontologies.

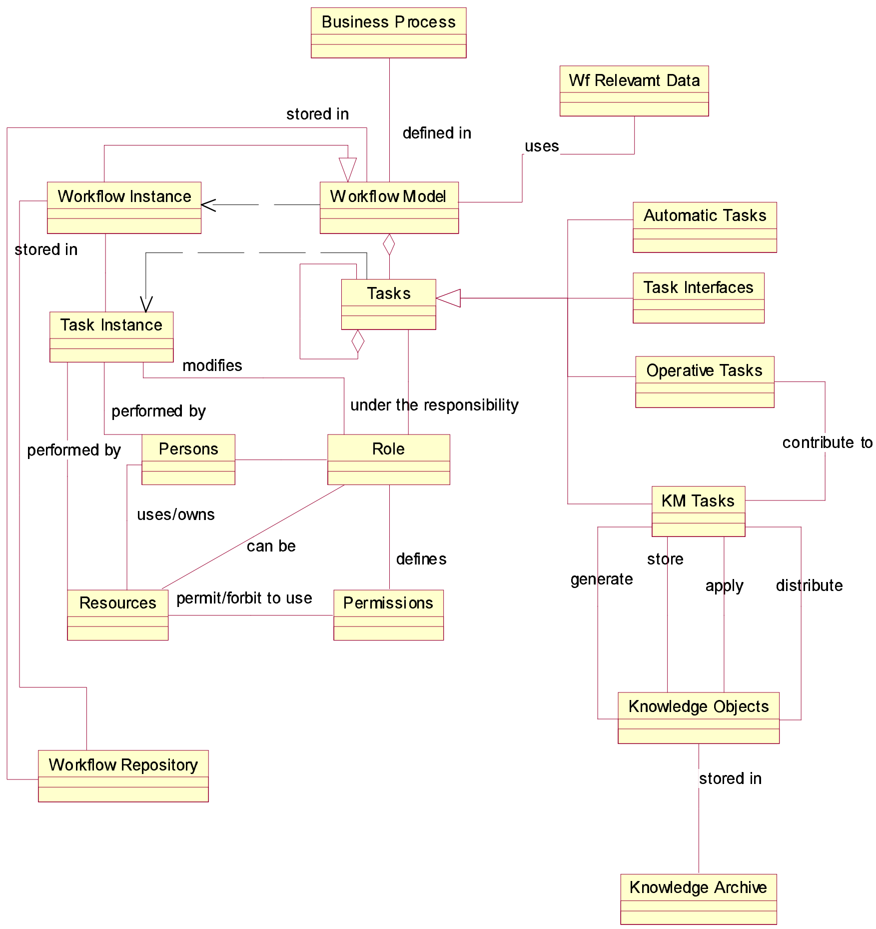


Figure 56: DECOR Workflow Metamodel Using UML Notation

Modelling the Knowledge Perspective.

As we already saw, there is a straightforward mapping between the typical ontological metamodel elements and the CognoVision data structures which is summarized in Table 28

IDEF 5 / Ontology wording	CognoVision implementation
Kinds / Concepts and Instances	Structure Elements
Characteristics / Attributes	Definable attributes
Relations	CV Links

Table 28: Ontology and CognoVision Terminology

Consequently, in order to operate economically, we did not design or link into the scenario a new, dedicated Ontology Modelling tool, but simply used the CognoVision Administrator tool for defining the required types of structure units, definable attributes and links required for reflecting the elucidated ontological structures.[113] If we have done this, we have created the not-process related part of our archive structures and can already insert document and other knowledge objects, categorized wrt. domain ontology structures.

In order to add the process / workflow model structures, we use our modelling method which will be introduced by means of examples taken from the IKA (Greek Social Security Institute) case study of DECOR.

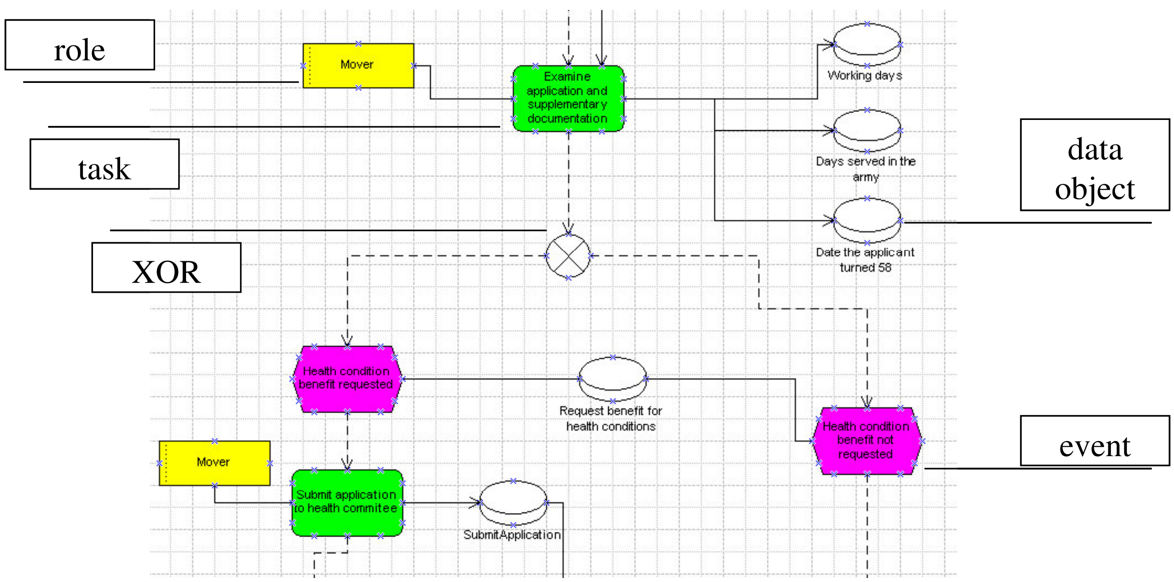


Figure 57: DECOR EPC[114]

---

[113] We have to keep in mind that DECOR was a research project, not product development. Hence, this approach was sufficient for a proof of concept. In a further productization of the DECOR solution, one would probably establish a link between well-defined, stepwise, tool-supported methods for Ontology Engineering (like Ontoknowledge with the Onto* suite of tools [Sure, 2003]) and would then – similar to our VISIO-CognoVision coupling – establish a dynamic link which directly reflects ontology modelling actions in CognoVision data structures. Or even better, one would build such an Ontology Engineering tool also as a VISIO modelling method, closely integrated with the earlier analysis steps and the process modelling. We briefly discuss this vision for a fully integrated, complete set of tools, at the end of this Chapter.

[114] The reader familiar with Business Process Modelling may notice in one of the following figures that we do

The DECOR extended Event-driven Process Chain.

Process logic modelled as Event-driven Process Chain

Tasks are connected with events using control flow elements (sequence, and, or, xor) forming an Event-driven Process Chain (EPC). EPCs are extended by links to other relevant entities. In this way, tasks can be connected (i) to input and output data for modelling the data flow between different tasks, (ii) to knowledge objects (documents, html pages, lessons learned) for modelling knowledge flow, and (iii) to resources for modelling the organizational perspective.

Process flow modelling primitives

In the DECOR EPC, control flow is modelled using sequences, splitters and joiners.[115] With the sequence flow element, it is possible to link two tasks sequentially. More interesting are the split-join constructions that allow a workflow path to split into multiple parallel branches. It can be specified that such parallel branches all have to be executed at the same time (and-split), that only one (xor-split) or some (or-split) of these branches have to be executed.

Figure 57 depicts part of the EPC for the IKA case. An XOR-connector is used to split the flow of work into two possible branches. During run-time the work flows in only one of the two branches, based on the value of a data object that has been

---

not always denote completely syntactically correct Event-Driven Process Chains. Hence we should call them DECOR EPC. We do not adhere to the EPC modelling rule that each activity must be preceded and followed by an event. The reasons are again pragmatic: we had to find an easy to understand and easy to implement way of defining executable process models. For process automatization however, the events produced by finishing a task are often directly reflected in value settings or changes of certain decision variables which are later on used for process automation. In order not to model this information duplicate, we decided to abandon the produced event after an activity in the cases where this effect of the activity is already clearly represented by a variable. So, in the DECOR EPC, an activity is not always followed by an event, but always followed by an event *or* finished by setting a variable value. Of course, for a refined implementation of the research prototype, this modelling anomaly should be removed.

[115] Again, it should be noted, that our project focus was not to design a perfect business-process model, but to prove that the combination with KM services is feasible and useful. Hence we included those process modelling constructs that were needed for our case studies. This should not be interpreted such that we would claim this to be the ultime process modelling language. In contrast, if we would have gone deeper into this topic, probably the need for some other process modelling primitives would occur. For example, the representation of timing constraints was really necessary and implemented for the CHUB case, not discussed in this thesis. Generalizing this idea, one could also think about inclusion of constraints over other objects under the control of – or accessible by – the workflow engine (i.e. variable values, e.g. for context variables). In CHUB, we had also to find an elegant way of expressing some “polling” behaviour when waiting for input coming from systems external to the DECOR software.

set earlier. The usage of data objects in the workflow model for controlling the flow of work is explained below.

Task interface

Interface task objects.

For the sake of modularization of models, in order to make them better overseeable and understandable, "Interface task" objects can be used to link together two different models or Visio pages (see Figure 58).

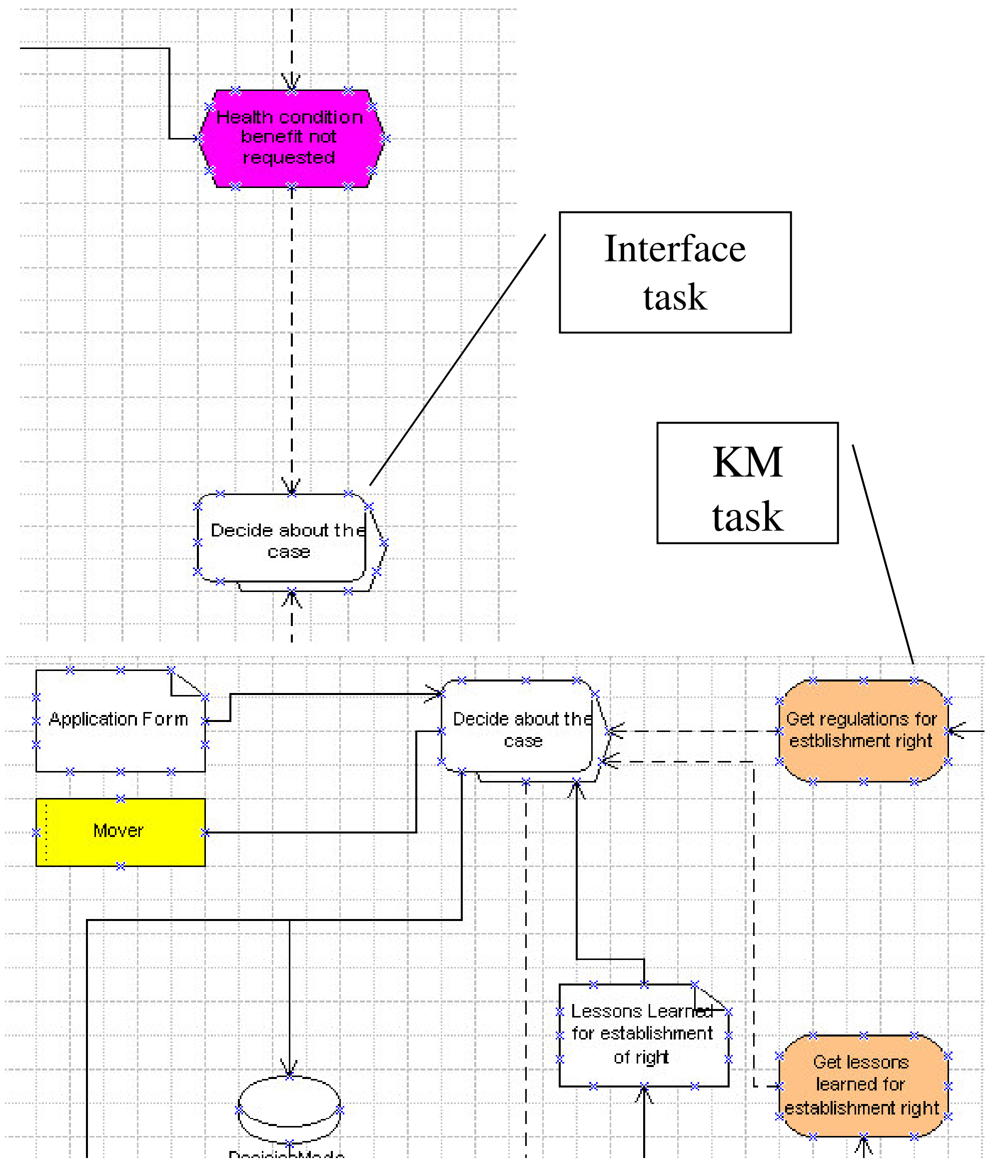


Figure 58: Using the "Interface task" Object

Modelling knowledge tasks into the business process.

Knowledge Management Tasks

In order to support the modelling of knowledge-intensive processes, we introduce an additional kind of tasks, the KM tasks or knowledge tasks. This kind of tasks is used for two reasons.

Automation of knowledge tasks

- Model the *automation* of some knowledge-oriented tasks, i.e. to offer active retrieval of information. In the IKA case, a typical example of such a knowledge task is the retrieval of regulations regarding the establishment of the right for a person to receive a full old age pension. Here, were access Mnemonic functions of the OMIS KBL.

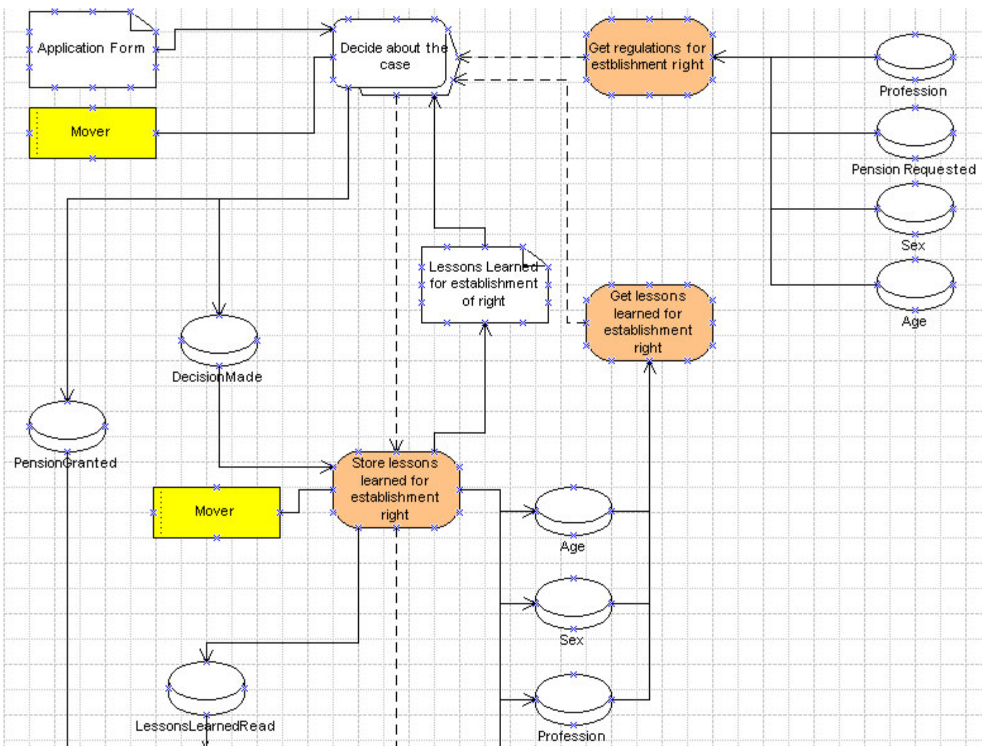


Figure 59: Using KM Tasks

Extend process using new KM Tasks

- *Extend* the business process by additional KM Tasks, like retrieval and storage of the lessons learned from each instance of the process: They are also automated, but add a new functionality to the model, whereas the information search mentioned above might have been done already before, but manually. In the IKA case, e.g., we added a Lessons Learned cycle to the process, where experience is generated during the enactment of a process, stored, (distributed), retrieved within the next process instantiation (and applied). These KM Tasks are explicitly modelled in the workflow model.

Automated retrieval tasks

KM Tasks are shown in Figure 59. The task at hand to find a decision about the applicant's request to receive a full old-age pension. We can see that this task is connected with three KM tasks: (1) "Get regulations for establishment right", (2) "Get lessons learned for establishment right" and (3) "Store lessons learned for establishment right". The link between the Normal task and the the first two KM Tasks is of type "contributes to" which means that they are not executed by the user (i.e. added in the user's worklist) but they are executed automatically, in order to offer proactive help for coming to a decision. During run-time, when the specific task is activated, the connected context-variables are used for triggering a search in the DECOR archive system for relevant regulations and lessons learned. The details of this search (i.e. the actual support request) is specified as an attribute value of the KM task – not shown directly in the visual process model.

Automatic storage

The third KM Task is done only partially automatic because the storage of lessons learned obviously requires some human intervention in order to reflect about the decision made and write the insights down (naturally, this is an optional task – normally, onyl very few cases will lead to a new lesson learned).[116]

[116] The automatic part mainly consists in using the associated context variables plus general properties of the actual process instance for automatically creating metadata which describe creation context.

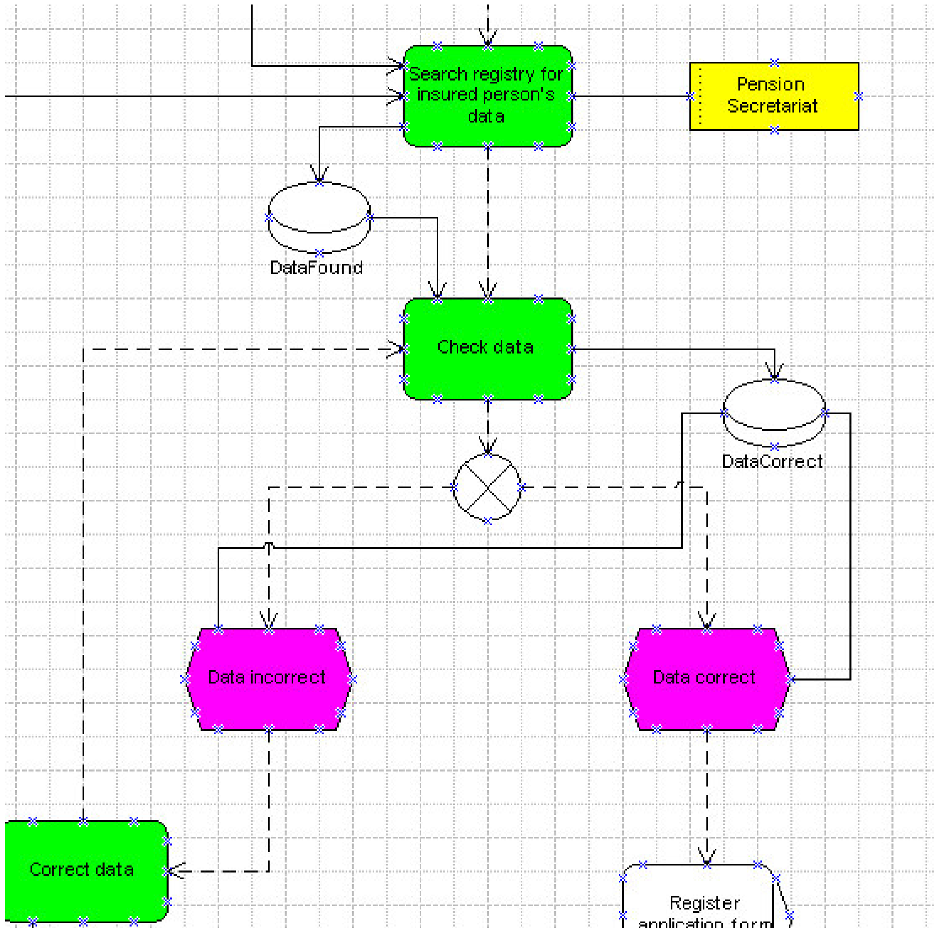


Figure 60: Data Objects for Modelling Control

Data flow.

The data flow within the business process is modelled by linking tasks with the data (depicted as data objects) that are used as an input and produced as an output of the task.

Data objects may be ...

Data objects represent either workflow-control variables for controlling the flow of work, or context variables used later in active retrieval / storage tasks. In Figure 60, one can see the usage of a data object for realizing control flow in the process.

workflow-control variables ...

The task "Search registry for insured person's data" is followed by the data object "DataFound" and will take the value "true" when the task is completed – which has to be the case before the task "check data" may be started.

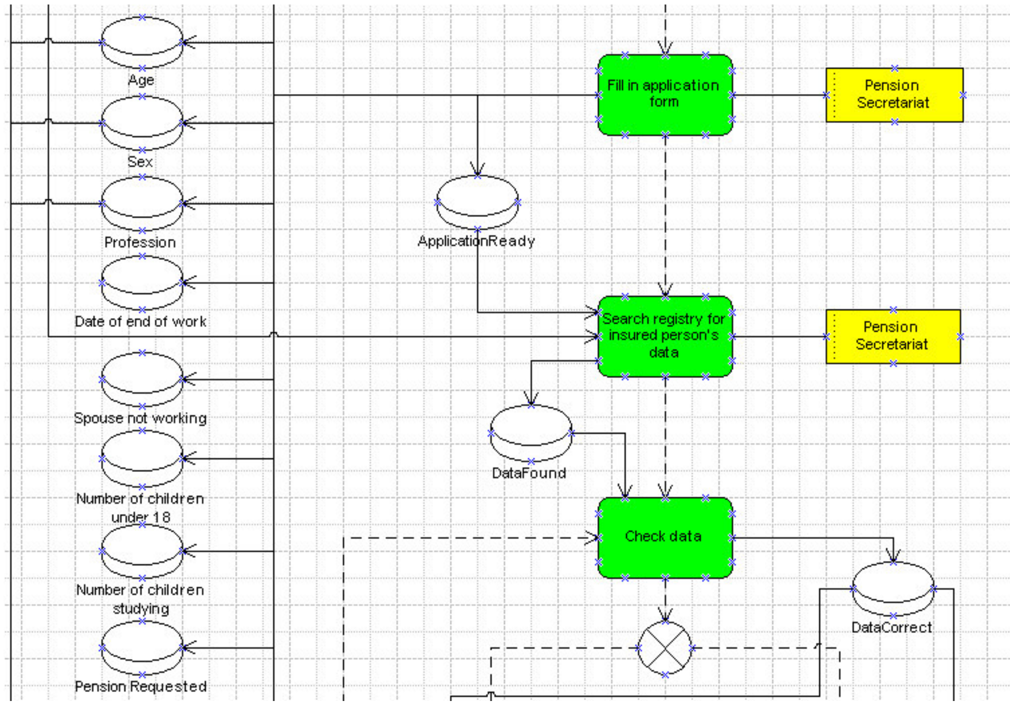


Figure 61: Using Context Variables

This kind of variables is also used whenever complex branching in the workflow is required. In the same picture (Figure 60), the data object "DataCorrect" is used to decide which branch of the XOR-connector should be followed. It takes its value from the task "check data" and serve as input to the events "Data incorrect" and "Data correct". If the value of the variable is "true" meaning that the data checked in the task are correct, then the branch where the "Data correct" event is located is followed. Otherwise, the user is led to the task "Correct data" and from them back to the task "Check data". This loop is continued until the value of the "DataCorrect" is set to "true" meaning that the user can proceed with the next task, i.e. "Register application form"

... or, context variables.

The usage of data objects as context variables is shown in Figure 61 and Figure 62 for setting and using them, respectively. During the task "Fill in application form" the user fills some data in a form. These data are transferred (manually, or by some software mechanism) into data objects and then can be used as input for the subsequent KM tasks "Get regulations for pension amount", "Store lessons learned for pension amount" and "Get lessons learned for pension amount". In those KM tasks, the respective variable values together constitute the context that will be used by the retrieval agent: For instance, for finding the regulations that apply in the current situation – regarding a specific type of profession, or a specific age of the applicant – which could be useful to know in the task "Calculate the pension amount".

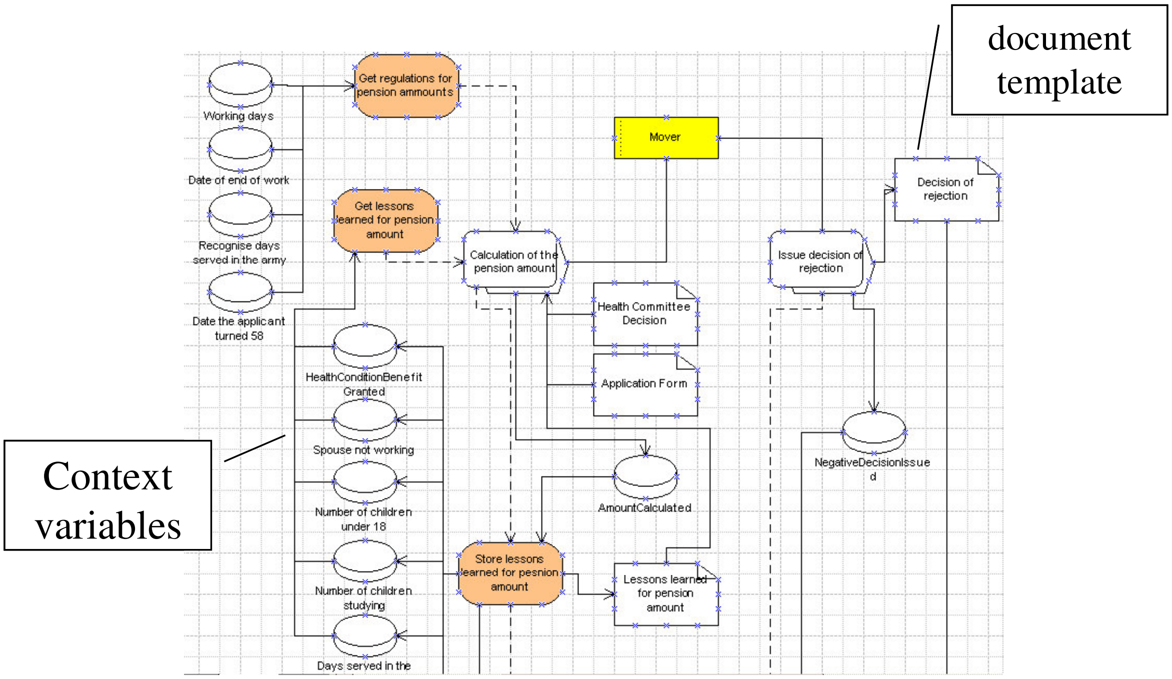


Figure 62: Active Retrieval Using Context Variables

Information flow through knowledge objects.

Knowledge Objects contain the information exchanged between tasks

The information flow during the process is modelled by linking taks with knowledge objects (documents, html pages, lessons learned). Tasks consume documents as input or produce them as an output. Context variables are used internally for describing attribute values of knowledge objects. In Figure 63, the knowledge object "Lessons learned for establishment right" is an output of the KM task "Store lessons learned for establishment right", and serves as an input for the Task "Decide about the case". The context variables "age", "sex", "profession" and "reason for the decision" (the last one cut off in the figure) are all attributes of the specific knowledge object, and they are used for the retrieval of lessons learned.

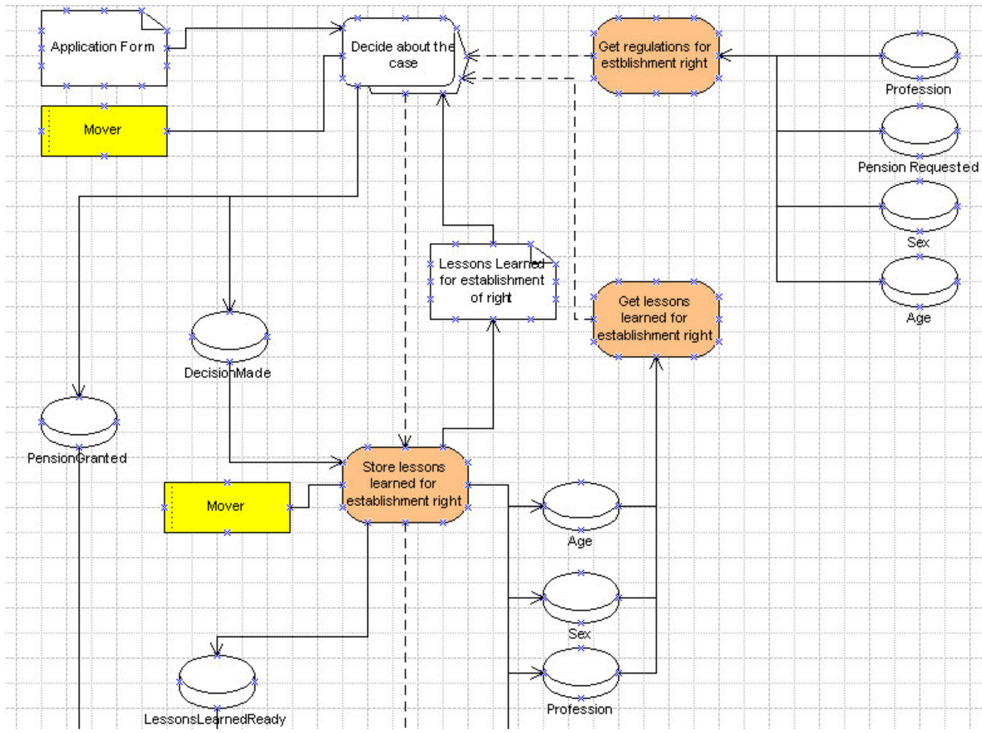


Figure 63: Knowledge Objects

Assigning roles to tasks.

Assigning roles

The last missing of the workflow modelling is to specify the roles that perform each task. As shown in Figure 64, this is done by linking the "Role" object to the "Task" object using the link "executes". These roles will be later associated with real persons in the responsibility diagram.[117]

[117]In this thesis, we do not show the DECOR organizational modelling (for details, please consult [Decor, 2002]). As said above, the responsibility diagram associates people with organizational roles. It is similar to the respective modelling facilities in tools like ADONIS. We did not implement the full complexity of a comprehensive organizational model as indicated in Subsection 3.1.3. However we would consider it useful to come to some standardization in this area, e.g., based upon a thorough analysis as presented by [Muehlen, 1999].

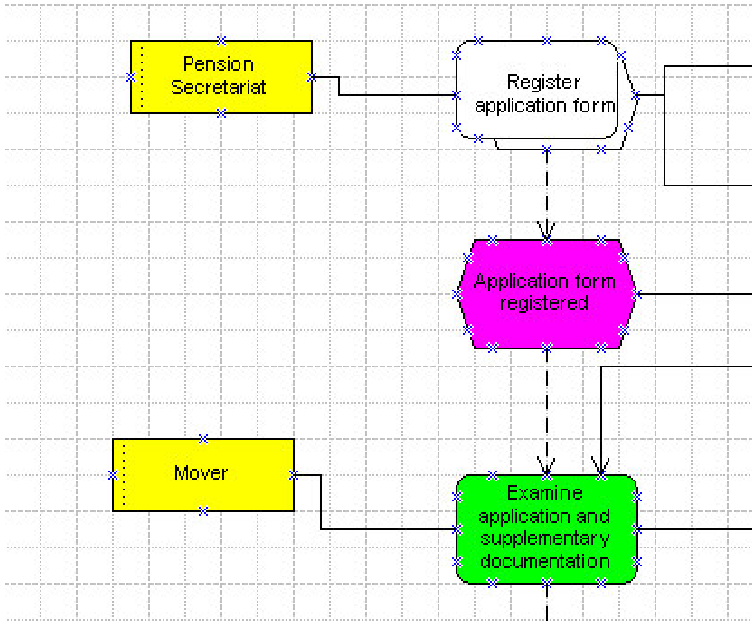


Figure 64: Assigning Roles to Tasks

Representation of workflow models in CognoVision.

HTML-display of the workflow model

As we mentioned already, modelling actions in VISIO are directly translated into corresponding data-structure manipulations within CognoVision. There, the modelled entities (tasks, data objects, information objects, links, responsibility diagram) are represented as Structure Elements, Knowledge Objects, Attributes, and Links. Since we also store the model as a whole in a particular type of CognoVision Knowledge Object, we can easily create a process-oriented knowledge archive (cp. 3.6.2), as follows: via the standard CognoVision functionalities, we can immediately browse through the lists of activities, their interrelationships within a process model, etc., and can also navigate to arbitrary supporting material which is linked in CognoVision to the respective structure elements. Furthermore, the HTML representation of the VISIO visualization can be used as an entry point such that all modelling elements in the diagram are an anchor equipped with a hyperlink which allows to directly jump to this structure element in the CognoVision browsing menus. In this way, we allow navigation through a knowledge network with process models as one possible structuring dimension, plus arbitrary other structuring dimensions, e.g., given by the domain ontology represented in CognoVision data structures.

Because it is possible to create links from other objects in the CognoVision database to shapes of the model, models can be connected to arbitrary other parts of the knowledge network in CognoVision. In this way, business process models and business process documentation can be connected. Models can be published in the enterprise Intranet and used as base for navigation through the business process documentation.

Finally, Figure 65 shows a VISIO process model in CognoVision, together with the internally created CognoVision objects that represent the workflow in the tool, and that can be used for knowledge organization.

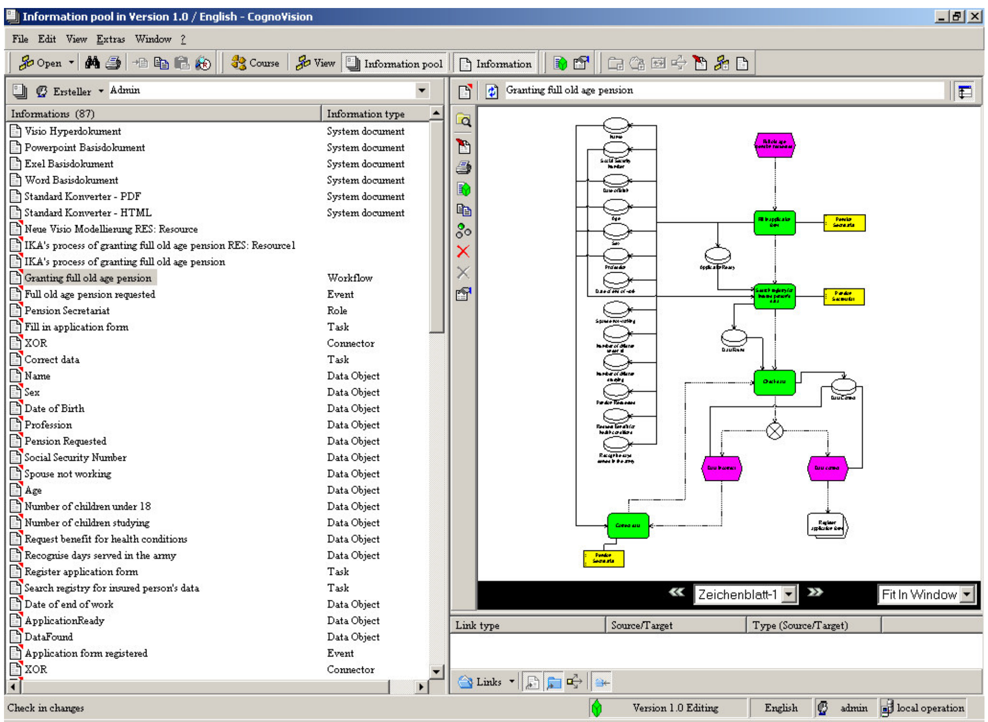


Figure 65: Navigating through a Workflow Model in CognoVision

4.4 The DECOR Smart Workflow Engine

The basic functionalities of the DECOR Workflow Engine[118] are, of course, almost identical to the KnowMore workflow functionalities. What was done in DECOR, was – besides a re-implementation using state-of-the-art software techniques – mainly twofold: on one hand, the whole interface and usage concepts were reviewed rigorously in order to provide a self-explaining tool without any understanding problems for non-IT professionals as end users. On the other hand, there was a deep implementation coupling with the CognoVision tool in oder to fully exploit the functionalities of the archive system in a powerful and efficient manner. Since these developments are not of extraordinary scientific interest, we go very quickly through tool functionalities, workflow implementation, and cooperation between workflow and archive system. The most interesting part is to get an idea of the way of working with the tool which will be given later when presenting the IKA case study in Subsection 4.5.1.

4.4.1 Functionalities of Workflow-Triggered Knowledge Delivery

The following services are available to the end user:

- List of available workflows
- List of open tasks for the user
- Treatment of task by the user

After selection of a given task, the user gets all necessary information to execute the task. This includes:

- The name of the task and the name of the overall workflow instance.
- If modelled, some explanatory text to the task

[118]Again, a word about the relationship between the DECOR WFE and this thesis: The DECOR WFE was – under the author's supervision, but technologically fully independent – implemented by Tino Sarodnik at DFKI, using DHC's interfaces and GUI. The author's role was mainly to gather case study requirements, define functionality of the WFE, review applicability in the pilot implementations, and prepare publications and documentation.

- A list of decision variables to be set by this task.
- The list of I/O documents to be used, written or edited in this task.
- The list of background knowledge found by the active retrieval services.

Figure 66: Simple Data Input With Background Information

Figure 66 (taken from the implementation of the PVG pilot, see Subsection 4.5.2) shows the representation of a task to be processed „Change Prozess initialisieren" (initialize change process). Here, only the main variables to describe the change process are put into the mask in the left half of the task window. As a help, in the right part of the window we can click a hyperlink to go directly to the standard operating procedure (SOP) describing what to regard and how to proceed in this working step. Figure 67 shows a slightly more difficult task, the decision about approval or rejection of a change request. Here, the document list at the right side contains both the input document (the URS) which belongs to the workflow and the SOP URS which provides some background knowledge.

The following functions are available to the system administrator:

- List of all terminated instances of workflow
- List of all active instances of workflow
- Protocol of active / terminated workflow instances

- Workflow control functions for the administrator

Figure 67: Decision Task with Input Document and Background Information

4.4.2 Architecture of the DECOR Workflow Engine

Figure 14 shows the architecture of the workflow integration to the DECOR Basic Archive System realized in CognoVision. Seen from the hardware point of view, the whole solution is implemented as a Client / Server application with a web client provided to the end user with the help of dynamic HTML. The web clients access to the CognoVision-Server machine which hosts the CognoVision V3.0 system used as DECOR Basic Archive System plus the DECOR specific workflow extensions. All functionalities are rooted in a relational database management system which guarantees scalability, data security, transaction services etc.

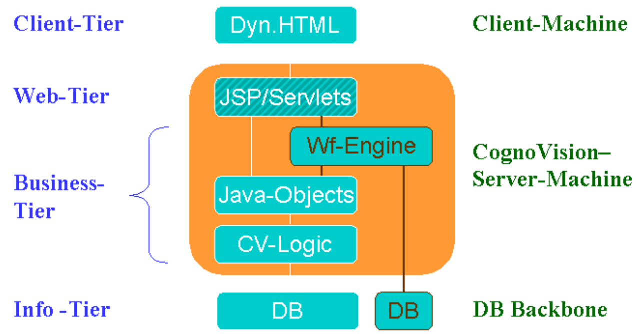


Figure 68: Architecture of DECOR Workflow Engine

The system is composed from a Business Tier which realizes the application logic, a Client and a Web Tier which realize the C / S functionalities in the Intra-/ Internet, and an Info Tier which realizes the data and information management of the system.

The DECOR extensions are seamlessly integrated with the software architecture of CognoVison, but nevertheless held strictly separate such that, principally, it is easy to decouple both components and reuse the DECOR WFE independent from DHC software. The architecture is based upon a business-logic layer accessible via Java-Objects by a set of Java Server Pages / Servlets which create the user interface. We implemented the WFE as a Java application which communicates with these CognoVision Java-Objects. The DECOR WFE has a data management of its own in an RDBMS, and directly extends / complements the DHC JSPs / Servlets by its own ones for creating the workflow-specific GUI elements and windows.

4.4.3 Cooperation between DECOR Basic Archive System and DECOR Workflow Engine

The overall cooperation of the several system modules can be described as follows:

1. The **user models a workflow** in Visio
    a. The dynamic link to CognoVision immediately creates the respective information objects.

  b. When closing the Visio model an HTML representation of the overall process is created and stored as a clickable process map together with the information object representing the whole workflow. The associated hyperlinks directly lead to the several information objects representing tasks, data objects, etc. with their attributes.

  c. Furthermore the information object representing the whole workflow is equipped with an XMLSchema representation of itself.

2. When entering the CognoVision screen from a web client, the workflow area offers an entry point to the personal space of tasks and processes.

3. When entering the personal workflow space, CognoVision is asked for all currently available workflow objects and gets a stream of XML representations of actual workflow models.

4. The user gets a list of workflow models (business processes) which may be started by him. To this end the DECOR WFE uses the responsibility diagram represented in the workflows plus the user management of CognoVision.

5. When **starting a workflow model**, a respective workflow instance is created in the WFE and represented as a set of interrelated tables in the WFE RDBMS. From now on the complete workflow instance information is stored in the RDBMS which guarantees recoveribility. Further, the RDBMS instance representation holds a complete audit repository for later process improvement, documentation, validation etc. In addition, at the moment of starting a new workflow instance, a corresponding folder in CognoVision is automatically created which gathers all documents and information objects created or changed in the process or its subprocesses.

6. Now the **DECOR WFE interprets the workflow process logic** and delegates the several tasks according to the decisions made by control variables and according to the modeled roles and responsibilities.

   Each task is sent to the individual worklist handler of the user/s which may execute it. Of course, when one user decides to work on the task, it is locked for the other users and disappears from their worklists.

7. When **a user executes a task**, a corresponding HTML representation in the web client is dynamically created which contains:

a. The task and process name plus (if modeled) remarks or explanations to this task.

b. Also in the right part **(task execution panel)** of the task window, there is a list of variables the values of which must be set in this task (e.g. an approval task determines whether some request shall be fulfilled and accordingly sets a decision variable to "yes" or "no"). The variables are presented in an appropriate manner, according to their value range (boolean variables or enumeration types as a pulldown menu with possible values, string variables as a text input field, etc.).

c. The right part of the window contains **(task support panel)** documents associated with these tasks. Basically there are two types of documents possible:

   i. Operative document – i.e. documents which belong to the modeled data flow of the business process because they are input or output document of this workflow. Since the DECOR WFE has full access to the CognoVision document management functions (see next subsection) this could also be a copy of a template if each instance of a specific workflow needs a filled in instance of its own some document template.

   ii. Supporting document – i.e. some knowledge object retrieved by the DECOR knowledge agent in order to give some helpful information to the task at hand.

      - This retrieval may depend on modeled context variables or may be executed by a complex knowledge-based retrieval agent. E.g., it could retrieve a legal regulation which applies just for people fulfilling some conditions specified in earlier stages of the workflow.

      - In the case of an automated task for retrieval this is started automatcially when entering the associated operative task and the retrieval results are inserted here in the task representation window. Such an automated retrieval could also be a more complicated process like asking the mindAccess document analysis system for finding an old risk assessment protocol produced in a case which was described by a similar text document as produced now.

- In the easiest way of a static background information always (for each different workflow instance) relevant in the same form, this might have also been modeled like an operative document (which is the case in the PVG example with the standard operation procedures SOP).

d. Depending on their role in the workflow the several documents can either be opened for reading or for editing. When a document is processed in a task, it is automatically inserted into the respective project / process folder in CognoVision (see above) such that there slowly grows a full representation of the process work which can be accessed later by the so-created task-oriented process / project portal.

This describes the most important aspects of the CognoVision-WFE inter-operation. Of course, the workflow administrator has some more specific rights and views which were already listed above. This software implements a seamless integration of basic workflow functionality with the DECOR Basic Archive System.

4.5 DECOR Case Studies

4.5.1 IKA Pilot[119]

IKA: The Greek Social Security Institute

The Greek Social Security Institute (IKA) is the largest insurance institution in Greece. Having as its primary purpose the protection of the insured persons, IKA offers an extensive range of services to them, like insurance, benefits, pensions and interstate social security. In 2001, IKA provided healthcare to 5.500.000 insured persons, including the members of their family. It paid out pensions to 1.000.000 pensioners, approximately. The Institute's income is derived from contributions of both workers and employers, and from governmental funding.

IKA business process supported with DECOR

The pension-granting process at IKA.

The business process that was examined and modelled within DECOR is the granting of full old age pension. The significance of the pension process lies in the large number of beneficiaries that, in the year 2001 amounted to 1.000.000 persons, increasing at an annual rate of 10%. The pension-granting process requires a deep knowledge of the relevant legislation; first for making the decision whether the insured person is entitled to receive a pension; and second for calculating the amount of pension.

Problematic situations in the pension-granting process

It is quite common that for one specific case, more that one legal regulation may be relevant; then, it is a matter of knowledge and experience to identify all these regulations and to choose the most appropriate one. If it is the case that the insured member can establish a pension right under more than one regulation, the different pension amounts are calculated, and the highest one is chosen.

The pension-granting process –as part of a normal administrative workflow – contains some central, knowledge- and document-intensive activities. These

[119] The presentation of the IKA case is a shortened and reworked version of the respective Section prepared by the author of this thesis for the DECOR Final Project Report [Decor, 2002]. It goes back to work in DECOR which was primarily undertaken by colleagues from ICCS and PLANET-EY, together with the IKA employees. Parts of the text have been published in a similar form in [Abecker et al., 2003; Papavassiliou et al., 2003].

knowledge-intensive activities, are often done with uncertainty (not all data are fully known), they are based on the experience of the relevant regulations the employees have (both regarding the time for coming to a decision and the quality of the result), and they are vital for the correct result of the process. Of course, the results of these activities must be legally checkable.

Starting a pension granting process

The process begins with the submission of the application form by the insured person and the collection of all the supplementary documentation, which constitutes the retirement folder. The retirement folder is submitted by the insured person to any of IKA's branches, and then it is forwarded to the one being responsible for acting upon it. The pension folder is being checked at the department of pensions or the department of payments. If it is not complete, a communication takes place in order to receive the documents that are required for the establishment of the pension right; this communication is between the department of pensions / department of payments and the insured member, or between them and other departments, or even with other branches.

Deciding about the entitlement to a pension

An insured person is entitled to pension when he/she fulfills the prerequisite conditions (e.g., minimum number of working days and age) for the specific type of pension and category to which he/she belongs. The decision regarding the entitlement to a pension is made on the basis of the employment and personal data of the insured person. This decision is based also on the current legal regulations, which are differentiated according to the pension type, the category of the insured person, and some other factors.

Having established that the minimum prerequisite conditions are met, a decision of approval is issued, which mentions all the information related to the granting and the calculation of the pension. If the insured person is not entitled to a pension, a decision of rejection is issued.

Applying the DECOR Business Knowledge Method.

After application of the Business Process Knowledge Method, the model of IKA's "Granting of full old age pension" business process was developed, enhanced with Knowledge Management tasks for the knowledge flow in the business process. Figure 69 depicts part of the model as it is presented to the user in CognoVision.

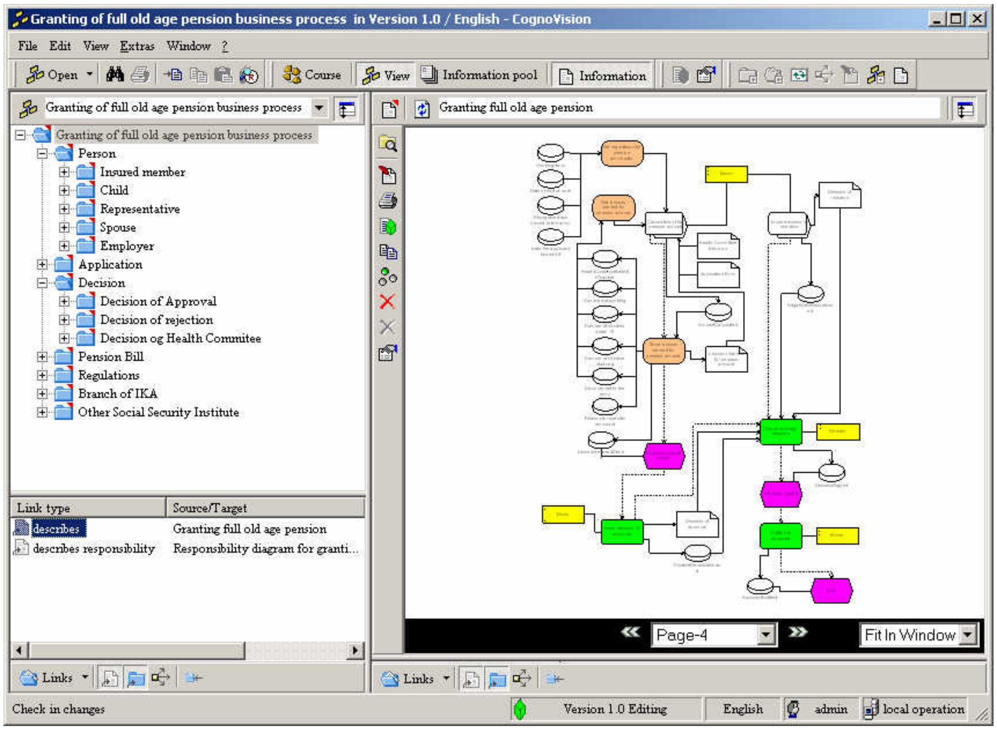


Figure 69: Part of IKA's Business Process Model

The respective domain ontology was relatively simple and (1) mainly connected people with that properties which were relevant for deciding upon their pension right, such as profession, years of working, etc., and (2) represented the different kinds of professions with their particularitites. Knowledge-intensive activities were found especially when deciding about the entitlement to a pension right and when calculating the pension amount.

Installing and testing the prototype.

Installation and initial testing of the IKA pilot

In order to test the DECOR tool for the "Granting of full old pension" business process of IKA, the following approach was adopted:

1. The tool was installed on an autonomous server PC of the IT research department of IKA in Athens.

2. An initial testing of the tool was performed by ICCS and PLANET-EY. This initial test involved running with the IKA pilot five past cases of insured members, i.e. cases which had already been examined by IKA. Thus, having

the respective decisions at hand, the system was especially tested whether it retrieved the relevant regulations. It is clear that the retrieval of similar lessons learned could not be tested because, at that point of time, there were not enough cases in the knowledge archive.

Training workshop with IKA personnel

3. Following the initial test, and after ensuring the proper operation of the prototype in terms of workflow execution of the business process, a training workshop with the IKA personnel was organised. The demonstration of the system involved first processing with the system two past cases by ICCS / PLANET-EY. After clarifying to the IKA personnel the way the system operates, three other past cases were processed with the system by the IKA personnel. ICCS/PLANET-EY were present in order to answer questions and give clarifications were needed.

Operation of the tool by IKA personnel

4. The next step was the operation of the system by IKA personnel with 15 additional *past cases* in order to fill in the archive and create an initial knowledge base with similar cases (Lessons Learned). The cases were carefully selected in order to be representative and contained at least one occupation category (e.g. construction workers, syndicalists), both sexes and spanned across different age ranges.

Testing the tool with new cases

5. Finally the system was tested again by the IKA personnel with 15 *new cases*. These cases were applications of insured members recently submitted to IKA for which no decision had yet been issued. During this phase indicative time measurements were taken in order to derive an initial assessment of the speed in executing the business process with the aid of the tool.

Quantitative results

After all, the following quantitative measurements for the effect of the DECOR tool were observed:

Criteria	**Reference measurement**	**With DECOR**
Number of decisions issued per day (in case all the respective documentation is available to the person examining the application in order to issue a decision)	2,4	4
Number of decisions issued per week against the number of submitted	21,86 %	43 %

applications per week		
Percentage of appeals to IKA's decision	10%	9 % (estimated)

Table 29: Some Figures about the IKA Case Evaluation

Qualitative results

At the qualitative level, interviews with IKA personnel yielded the following expected benefits:

- Possible use of the tool for *training purposes* for new employees
- More *consistent treatment of cases*, because all employees can access the same regulations
- *Faster dissemination of new regulations*, best practices, or other new information
- General benefits from *process automation* instead of manual document handling

In the Appendix in Chapter 7, we use the IKA business process as a running example for demonstrating the use of the DECOR system.

4.5.2 The PVG Case Study[120]

The PVG change management process.

The Plasmaverarbeitungsgesellschaft mbH, Springe is a subsidiary of the German Red Cross dealing with blood donours, blood plasma processing, blood products, etc.

PVG Change Management Process

In the PVG case we are handling the business process of change management for the validated SAP R/3 system of PVG. The process of change management starts if an user of the SAP R/3 system has a change request. These changes can be of the following types: software development, customizing or changes in the system master data. The change request is classified depending on the affected business processes as critical or not critical and the change is associated to on of the three

[120] The PVG case was mainly analysed, modelled and coached by DHC employees. The author of this thesis supported the analysis and evaluation and mainly wrote the presentation of the case in [Herterich et al., 2003] which is a prior, shorter version of this Section.

categories: great change, standard change or small change. Depending on the risk classification and the change category one of the following four procedures for the execution of the change has to be applied: revalidation, standard procedure, simplified procedure, uncritical procedure.

Major process steps

Overall, the process consists of the following main steps: (1) Risk analysis; (2) Specification; (3) Implementation; (4) Functional test; (5) Acceptance test; and (6) Release. Depending on the concrete procedure (see above) some steps can be omitted.

Involved roles

The following organizational roles are involved in the process: (i) the user of the SAP system; (ii) the Quality Manager; (iii) the modul administrator for the SAP system; (iv) a consultant or software developer; (v) the administrator of the SAP system; and (vi) the SAP project manager.

Figure 70 presents a part of the PVG change management process, shown in the CognoVision tool.

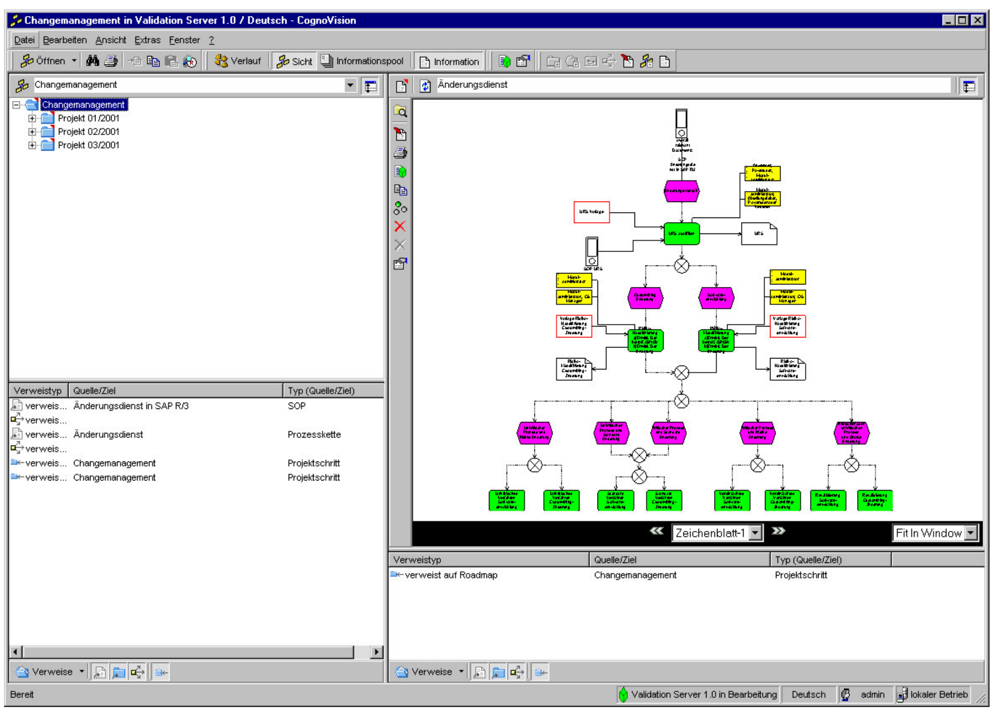


Figure 70: Visualization of Part of PVG's Change Management Process

Characteristics of SAP change management at PVG.

With respect to its KM support potential, the SAP change management possesses the following remarkable characteristics:

1. It is critical to ensure the continuously validated state of PVG's production which is a key criterion for quality in the pharmazeutical area.
   mission-critical process
2. It involves many tasks, is highly document-oriented, and involves several people in different organizational roles.
   typical "workflow candidate"
3. Changes must be documented according to national and international rules.
   documentation need
4. The change process must follow provably the national and international rules.
   typical "workflow candidate"
5. Correct and effective execution requires for some tasks manifold kinds of background knowledge which today is often neglected (like standard operating procedures, templates, legal information, SAP background knowledge, etc.).
   knowlede-intensive process
6. The knowledge level of different people in the process, as well as between different people enacting the same task in different process instances, may differ considerably.
   knowledge sharing opportunities
7. The continuous improvement of validation quality and validation efficiency is an ongoing task.
   knowledge management need

In this list, while the first argument justifies the importance of the process chosen, the items 2 to 4 ask for workflow support, ant items 5 to 7 demand a KM solution as laid out by the DECOR approach.

Since all software components operational in PVG must be in a validated state achieved by long-term systematic testing, the following strategy was pursued to install and test the DECOR solution and come as far as possible towards an operational system:

- To define the baseline metrics for improving the quality of PVG change management, DHC performed an audit assessing the percentage of correctly performed and documented changes.

- The process-oriented knowledge archive for PVG was installed in PVG in Spring 2002. Since then, all changes in the PVG system were documented using this system.

Measurements before DECOR

Baseline metrics. During the initial quality audit by DHC for assessing the quality of the current change management process, the documentation discipline and the resulting documentation quality were evaluated. We cannot present detailed figures here. However, it can be said that a significant improvement potential was identified. For instance, in only 10% of the considered cases a test documentation was delivered. Further it turned out that the quality of the technical realization of a software change was not homogeneous. For instance, the test behaviour of different users differed considerably, also the documentation behaviour, and the implementation efficiency.

Expectations. With the above sketched status in mind, PVG's head of Quality Assurance expressed the following general expectations for the introduction of the DECOR system:

- Quicker and easier workflow
- Changes will be done completely in compliance with defined procedures
- Changes will be documented completely
- Improved planning and dating of work

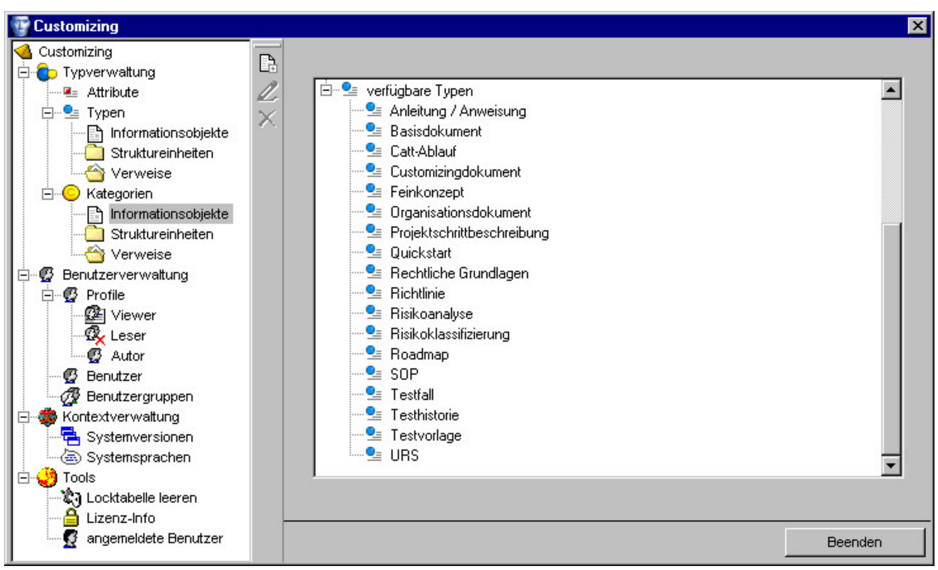


Figure 71: Types of Knowledge Objects in the PVG Archive

The PVG Process-Oriented Knowledge Archive.

The knowledge archive installed at PVG contains 780 documents of 15 different types of Knowledge Objects (see Figure 71). They are organized according to the structure of the change process, the structure of the SAP system considered, and thePVG organizational units, ... In detail, the following views are available (which represent – in terms of our ontological vocabulary – top-level concepts of domain and enterprise ontology):

Size of background knowledge base in the "Validation Server" knowledge archive

Available views

- Customizing documentation
- Procedure model prospective validation
- Procedure model retrospective validation
- Change management
- SAP R/3 menu structure
- Business processes
- Validation
- Project organisation

In total, we have 44 Structure Elements in the respective ontologies. This shows that – although a considerable number of documents was found to be useful as validation background knowledge – the organizing structures are relatively lean, in order to give to the end user some easy to use entry point, and not to overburden him with over-complex models. On the other hand, these few basic elements could be connected by a relatively high number of links (see Figure 72). This might give an idea of the high degree of interrelatedness within the PVG knowledge archive. This is one reason that such a system – just used as a process-oriented knowledge archive – can be problematic, since the user might get lost in the knowledge network. Here, we can expect the usefulness of process-oriented methods which guide context-sensitively to those Knowledge Objects which are currently relevant.

The underlying ontology has few nodes, but is highly interconnected at the instance level

The finally running test installation was managed by PVG's Quality Manager and used by about 20 active users, with a quite different individual usage profile. Some were regular users, others were very seldom confronted with the system, or with

SAP changes in general. In the first few months since the installation of the archive system, about 3300 document accesses could be counted.

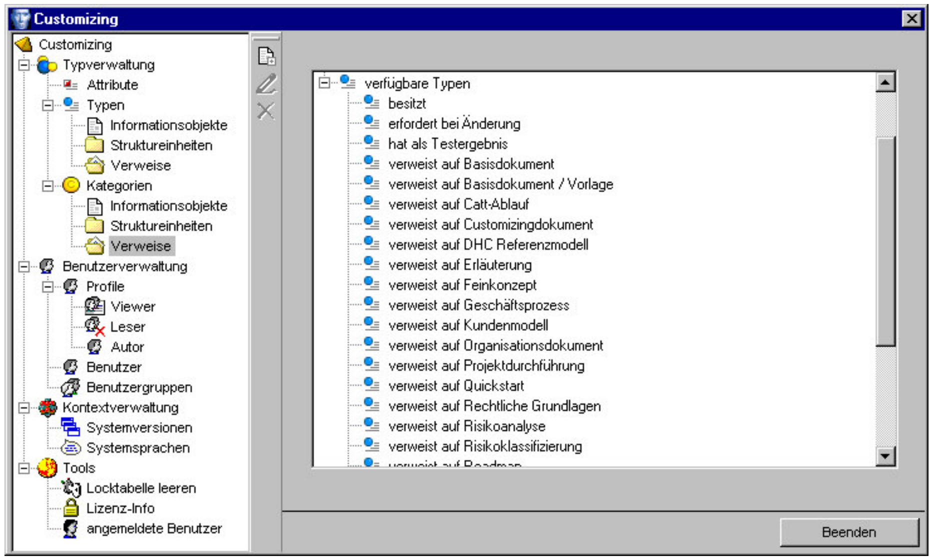


Figure 72: Types of Links in the PVG Archive

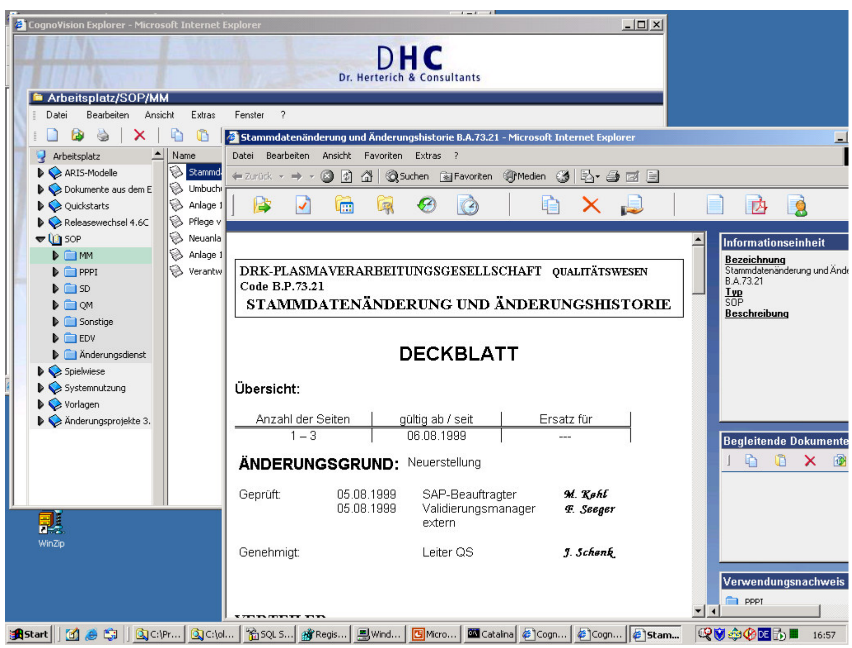


Figure 73: A PVG Sample Document (foreground) and the Underlying Knowledge Network (background)

PVG workflow enactment.

The PVG change management workflow contains about 55 tasks performed by 10-15 different people (or, organizational roles). Its major purpose is to ensure the compliance with all official regulations regarding document flow, logics of approval, and documentation. The major KM tasks, which extend the conventional workflow tasks, concern:

KM tasks in the PVG workflow

- Access to SOP's and document templates for a given task (static task context)
- Access to specific background information or earlier, similar changes for the given change process instance (dynamic task context: department which requests a change, SAP module affected, risk class, change class, change type)
- Automatic creation of a project folder per process instantiations which gathers all documents created during the process enactment, and automatic establishment of the required links between documents

Figure 74: Task List Handler of DECOR Workflow Engine

Figure 74 shows the task broser for one of the early tasks in the workflow, namely the classification of a requested change; the end user has to decide about the relevant data characterizing the given change request: department affected, SAP module, change class, change type. These variable values can be set in the left part of the window. As supporting material, the user may access the Standard Operating Procedure (SOP) for change management in the PVG R/3 system, which is offered as a hyperlink in the right part of the window.

Figure 75 shows the task "Check URS" which is about approval of a user requirement specification (URS). Of course, besides the access to the respective URS document as an operational workflow document, we have also access to the change management SOP as background knowledge in the document browser at the right part of the window.

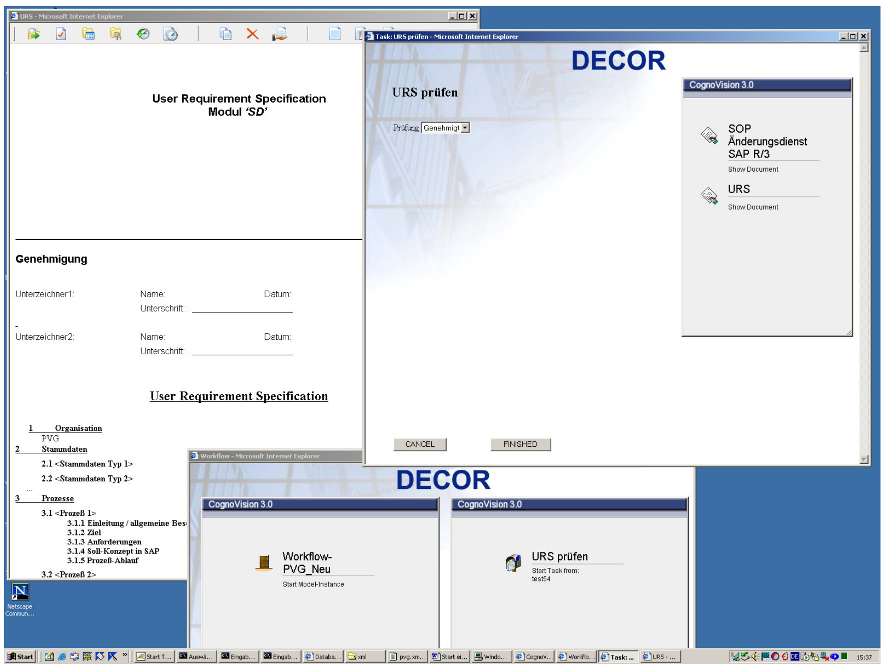


Figure 75: PVG Workflow Task "Check User Requirement Specification"

User Experiences and Reactions.

Usage experiences

For a serious quantitative evaluation, the PVG system was not tested long enough within the DECOR project. Nevertheless, the first impressions of the PVG Quality Assurance Department can be summarized: In general, two major user classes can be identified, where for each of them specific argumentations speak for the DECOR solution:

Power users can be monitored better

- For experienced users who work often with change processes (in particular, the SAP programmers) the biggest problem – seen from the Quality

Management perspective – seems to be that they often do not respect the regulations. Here the workflow enactment shall be a gentle means to "force" people to work in compliance with the rules. First, because it is easier to respect the procedures when guided through the automated workflow. Second, because leaving the official way means creating a "dengling process instance" which will be identified with the workflow administrator tool and examined by the QM manager.

  - Please note that in this case the best context-sensitive search is required for achieving user acceptance, because only a really excellent functionality can convince an expert user to do things where he thinks they hinder him doing the daily operative business.
  - As a corrolary we can derive the importance of continuously measured system performance (not only during a research project, but also in daily business) to demonstrate the usefulness.

- On the other hand, casual users in the change process normally have problems to overview its complexity, to know the regulations, and to follow the rules. Here, the system shall help them by giving appropriate guidance and background knowledge.

  Seldom users are better supported, guided, and helped

  - Therefor, *active* knowledge retrieval is crucial, since the casual user does not know what to search. Further it is a clear advantage to have a well-organized archive of networked information sources which allow a content-oriented browsing through the material.

Moreover, PVG's Quality Manager reported the following experiences from the first usage months:

- The integration of workflow and archive seems intuitive and useful. A reason for the usefulness seems to be that many people tend to think process--oriented step by step, and seldom in a systemic manner having in mind the whole picture with complex interrelationships. This problem can be ameliorated by working along the prescribed process, by having access to the relevant documentation, and by having the relevant links and relationships represented in the archive.

  The combination of process and content helps to achieve a holistic views

Expected performance improvements

- PVG estimated that the system reduces the time for a change by 10-15%. Of course, the gain will be smaller in complex changes where the major part of the elapsed time is consumed by the programming work for implementing the change.

Possible further extensions

- In this last case (complex changes, which do not profit much form pure automation of document and information flow) a much more important gain of change efficiency could be achieved a similarity-based search for older change documentations from similar situations. However, this is subject to future work.

  Anyway, it is clear that in such a highly sensitive area, the process throughput is not the main optimization criteria, but (provable) quality.

General benefits

After all, the combination of workflow and process-oriented archive proved to be worthful. Besides the concrete improvements identified, the major general benefit of systems as we sketched them, is that they help to exploit existing knowledge sources in a more efficient and more consistent way throughout the whole organization, and that they allow also faster distribution of changes of procedures, and of background regulations, as well as advice for enacting these procedures.

The PVG Process-Oriented Knowledge Archive is still in operational use and should be examined in a long-term study.

4.6 Related Work

There is a number of recent approaches which have some relevance to the work presented in this Chapter.

GPO-WM

Probably one of the first method-driven and tool-suported approaches to Business-Process Oriented Knowledge Management has been developed by Heisig and colleagues at Fraunhofer IPK, Berlin (cp. [Heisig, 2001; Mertins et al., 2003]) with their **GPO-WM® method and tool.** As they run successfully BPOKM projects since a number of years, they have obviously an overall KM project methodology. Their knowledge-oriented process and activity analysis provided some elements to our DECOR method. In particular, the distinction into demand-driven perspective and support-driven perspective, going through the steps of the whole knowledge lifecylce in order to identify and close gaps, goes back to GPO-WM®. They have an activity-based, object-oriented business-process modelling tool, the MO²GO tool for Integrated Enterprise Modelling. Further, IPK promotes the idea of systematically implanting best or good practices in business processes. To our knowledge, the IPK approach is not directly oriented towards workflow automation. Further, we are not aware of an explicit, sophisticated, treatment of evolution aspects in their approach (changing information needs, systematic content update, evolving system performance).

ARIS

In contrast to the IPK approach which is explicitly dedicated and oriented towards the particularities of Knowledge Management, nowadays available extensions of widespread **Business Process Modelling tool suites, such as ARIS,** are often not fully convincing in the way they treat knowledge. There are knowledge-oriented extensions since a relatively long time (cp. [Allweyer, 1998b]), such as:

- *new object types* “knowledge category”(“Wissenskategorie” – roughly, a domain ontology concept) and “documented knowledge” (“Dokumentier-tes Wissen” – roughly, a knowledge object type modelled in the informa-tion ontology); or
- *new model types* “knowledge structure diagram” (“Wissensstrukturdia-gramm” – sort of a simple combination of domain ontology and KDL

which arranges knowledge categories and document knowledge in a simple manner) and the "knowledge map" ("Wissenslandkarte" – associates people with competency profiles).

Like [Remus, 2002], we observe that the proposals how to treat knowledge in those appraoches (e.g., by [Hagemeyer & Rolles, 1998]), typically do not take into account appropriately the specific properties of knowledge (actuality, difficulty of classification, etc), and assume a too static understanding. Of course, there is also no explicit, strong link to workflow automation.

PROMOTE

A convincing combination of both approaches, the IPK philosophy and the ARIS-like, traditional BPM modelling, has been developed in the **PROMOTE project, method and tool** which are built upon the ADONIS® BPM suite [Hinkelmann et al., 2002; Woitsch & Karagiannis, 2003]. PROMOTE starts from similar modelling possibilities as just discussed for ARIS, but adds a more dedicated knowledge-oriented analysis. In particular, they introduced what I called the process-link perspective in Subsection 4.3.1, they offered a more expressive way of knowledge categorization – based upon knowledge networks represented as topic maps – and they emphasized the explicit modelling of KM processes. There is not a strong emphasis on workflow automation, however, sometimes they mention the possibility of powerful knowledge processing tools, e.g., for retrieval. In [Palkovits et al., 2003], the application of BPOKM methods to the e-Government scenario is discussed, a combination that we also consider highly promising (cp. [Abecker & Mentzas, 2001]).

Thiesse's approach

Thiesse gives in his thesis [Thiesse, 2001] an excellent overview of existing BPM methodologies and project methodologies, as well as KM particularities like modelling and analysis of knowledge-intensive processes. From that, he develops a **project management approach for running process-oriented KM projects.** As such, this could be an interesting input for further analysis with regard to the question whether our CommonKADS-driven "outer loop" for BPOKM projects could be improved, adapted, or extended. However, this would need more practical experience with real-world projects. Thiesse's approach is not tool-supported (actually, it could be run with arbitrary modelling tools), and it does not take into account newer developments in knowledge modelling as we do with our ontological middle layer. Hence, e.g., Information or Domain Ontology development could not be supported by his method.

To sum up, all discussed approaches which are stemming from the BPM/BPR world and were extended or adapted towards KM. DECOR seems to be the only approach that is more technology-driven and added a project and a process-modelling methodology later. Typically, the BPM-inspired approaches have reasonable project methodologies and modelling approaches; often, they do not emphasize workflow automation aspects or even more advanced technological innovations (such as system supported knowledge evolution); none of them focusses on ontology-based modelling of knowledge sources and knowledge content (information and domain ontologies).

4.7 Summary

To sum up the major contributions of the work presented in this Chapter, let us think back to the KnowMore OMIS prototype and architecture presented in Chapters 2 and 3, and let us have a look at Figure 76. While the KnowMore project and the respective parts of this thesis delivered a framework, a prototype software, and a proof-of-concept with some examples, the DECOR work extended this basis in two directions:

On one hand, **we developed one of the first tool-supported methods for running Business-Process Oriented Knowledge Management projects.** Appearantly, it is the first method worldwide to build extensively on an ontology-based middleware, that is closely coupled to a dedicated workflow automation, and that takes care for dynamic process context. If one takes the documentation of "surrounding" method elements from CommonKADS and for Ontology Engineering (IDEF) plus the DECOR project documentation and the Chapter 4 of this thesis (which considerably extended and clarified the identification and analysis steps of the original DECOR method by incorporating ideas from GPO-WM and PRMOTE), it is probably also the best documented method in this area (regarding freely accessible documentation).

On the other hand, we **technically consolidated the KnowMore status** by mapping a significant part of the KDL to the commercial CognoVision tool and by implementing a stable and comfortable Workflow Engine which communicated with CognoVision. This showed that innovative OMIS approaches do not necessarily require "esoteric" special software, but can be realized interoperable with existing tools and solutions.

Modelling tools, modelling method, and runtime enviroment together consitute the first *total solution* for BPOKM.

In the **DECOR case studies**, we gathered empirical experience with the practical feasibility of BPOKM approaches. We learned about potential benefits (such as situated, contextualized learning in the process; fostering knowledge sharing by case-driven transfer of lessons learned; shorter time for information search;

provision of formerly not known information; direct roll-out of new information etc.), but also limitations or risks (availability of technology; costs for system design and introduction; training efforts; danger of relying too much on the system; danger of resistance against prescribed work procedures; etc.). Altogether, the combination of BPM and KM methods is obviously promising and might lift a planned project over the hurdle of acceptance, because of the synergies (each stand-alone project might not pay-off, but with shared costs and double benefits, the situation looks better). Further, the process focus was in our experience a crucial plus for the KM aspects, since otherwise, end users simply would not have understood what we want. However, coming from the processes, we pick them up at their daily work.

Nevertheless, there is still a lot of interesting open questions and open issues for future work, which are discussed later. For instance, our understanding of the economics of BPOKM solutions (cost-benefit ratio) is still insufficient. This includes also – or, in particular – long-term effects with respect to potentially raising level of individual experience, influence on communication and sharing behaviour, maybe resistance to change, influence on knowledge sharing culture in general, etc.

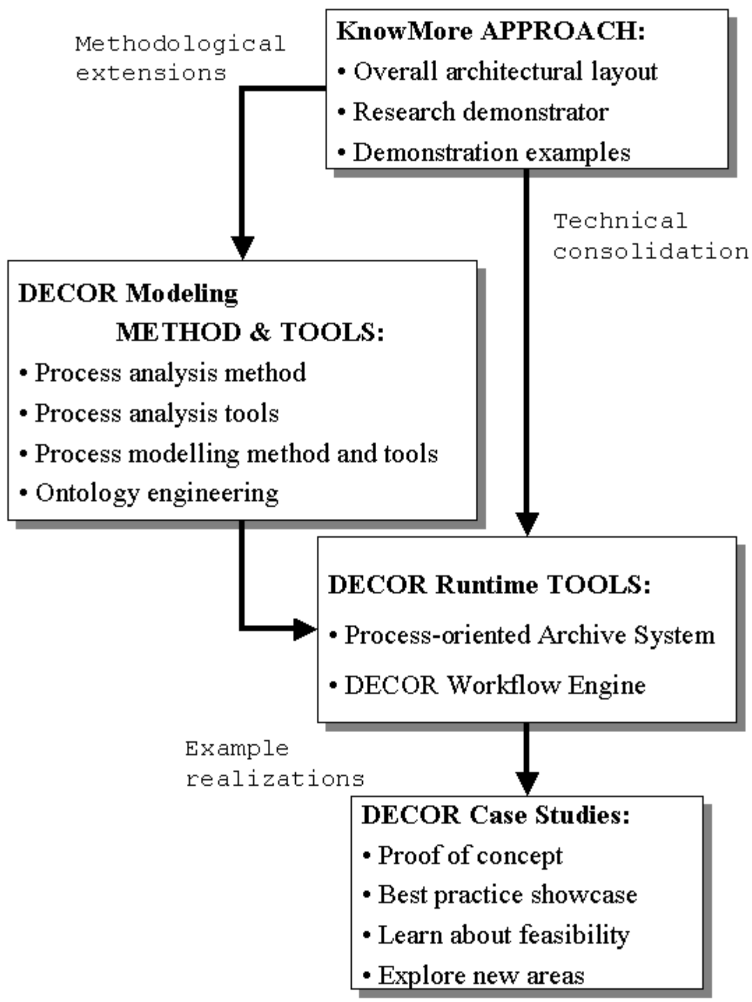


Figure 76: From KnowMore to DECOR

5 Derived Research Topics

The problem is never how to get new, innovative thoughts into your mind, but how to get the old ones out!
Dee Hock (VISA International)

Abstract: The work presented in this thesis was done over an exceptionally long period. It is normal that in this timeframe, a number of other promising ideas emerged which could not be worked on within the main stream of work or that did not fit into the main argumentation line of this thesis. Nevertheless, creating and cultivating such ideas was one of the author's major contributions to the scientific community over the last couple of years. Hence we would like to introduce three major areas of such "borderline work" even if they did not lead yet to final results and if they were not at the core of the author's own technical work. These areas are:

- The area of **Knowledge Trading**, i.e. online trading of knowledge objects or knowledge descriptions for knowledge-creating services. Both are relying on an extensive Information Ontology (see Section 5.1).
- The idea of **Agent-Mediated Knowledge Management**, i.e. of rigorously realizing a full OMIS setting by intelligent agent technology (see Section 5.2).
- The topic of **Weakly-Structured Workflow** which is based on the observation that knowledge work is seldom done in a way which could be easily supported by traditional workflow automation approaches (see Section 5.3).

Preamble: *The idea of Knowledge Trading came from Prof. Grigoris Mentzas. We jointly got the European RTD project INKASS (Intelligent Knowledge Asset Sharing and Trading, cp. [Apostolou et al., 2002; Apostolou et al., 2004]) accepted where the author of this thesis mainly wrote the workplan for and worked in for some months, together with*

Christian Reuschling and Sylvio Tabor. A central point for the realization of the INKASS workplan is the rich description of knowledge products on the basis of an Information Ontology. It was the author's ambition to elaborate this INKASS Information Ontology in such a way that it could serve as (at least as starting point for) a reference ontology. To come to this stage, was a hard piece of work – done in very close collaboration with Sylvio Tabor who – with an admirable, albeit sometimes exhausting ☺, pertinacity – challenged the design and every single detail of the ontology until we had the form presented here. Excerpts of this work have been published in [Abecker et al., 2003b]. The overall structure of the Information Ontology was proposed by colleagues from mcm Institute of Hochschule St. Gallen, Switzerland (cp. [Maass et al., 2003]). The topic of Agent-Mediated Knowledge Management goes back to the project proposal for the German bmb+f project FRODO (A Framework for Distributed Organizational Memory) that was designed and written – regarding this topic – more or less exclusively by the author of this thesis (cp. [Abecker et al., 2003; Elst et al., 2004a]). During the run of FRODO, the topic of agent-based workflow was taken up by Sven Schwarz and Heiko Maus, whereas the topic of agent-based Ontology Societies was defined and elaborated by Ludger van Elst. I enjoyed much to work with Ludger who interpenetrated the topics of Distributed OMIS and Ontology Agent Societies better than I would have been able to do it. This work is partially documented in [Elst & Abecker, 2002a; Elst & Abecker, 2002b]. Together with Ludger and Virginia Dignum (University of Utrecht), we coined the term Agent-Mediated Knowledge Management and hope to keep the momentum with this research direction (cp. [Elst et al., 2004b; Elst & Abecker, 2004]). Regarding Weakly-Structured Workflows: This topic arose from the very begin of the author's work on the topic of „knowledge work". Going back to sources from Horst Rittel, we found that conventional workflow systems do not offer appropriate means for supporting knowledge work. This topic was partially taken up by Sven Schwarzand Heiko Maus in the Frodo project.

5.1 Knowledge Trading

The fundamental paradox of information retrieval:
"The need to describe that which you do not know in order to find it"

Roland Hjerrpe

Motivation of the topic.

We observe two big trends in business and commerce in the recent years:

Economy asks increasingly for (ad-hoc) coalitions of small and flexible partners

- On one hand, there is a trend to integrate virtual supply chains, so transcending the extended enterprise towards huge, completely virtual value-creation chains.
- On the other hand, there is a – somehow corresponding fragmentation of enterprises – which might end up in small, dynamic, independent service-providing units.

Both trends together lead to opportunities and to challenges with regard to highly-flexible, adaptive, short- or medium-term business coalitions.

However, this ongoing process of fragmentation and dissolution of traditional organization forms, structures, and processes, also introduces new problems, especially in the area of organizational knowledge creation, retention, and reuse. Unfortunately, it's just this Knowledge Management area which gains ever increasing importance in a time of shorter product lifecycles, heavy competition, knowledge-intensive service and high-tech markets, and continuous shifts in the business environments (due to new technologies, changing customer interests etc.). As a reaction, the last decade was faced with a tremendous business interest in systematic and effective approaches for managing knowledge as a corporate resource, this thesis being an outcome of which.

However, although these efforts in Knowledge Management research and developments showed remarkable success, at least three critical remarks can be made:

Intra-organizational KM does not exploit all opportunities, in particular for small and medium organizations

1. Successful, holistic KM initiatives with measurable results, are still pretty much restricted to big, multinational trusts. They are not yet a widespread reality for small and medium enterprises (SME's).
2. The longer the above-mentioned fragmentation trends will continue, the more difficult the KM problem will become.
3. Even if *intra*-organizational KM initiatives produce excellent benefits, they do not exploit the higher potential hidden in *inter*-organizational knowledge networking, which aims at the combination and leveraging of complementary skills, knowledge, and experience, and which fosters effective knowledge-creation which often happens "at the borderline".

Knowledge trading combines e-Commerce and knowledge management

It seems a promising answer to the trends above, to combine E-Commerce and Knowledge Management, thus creating the new research topic of **Knowledge Trading.**

Knowledge Trading is not only a logical consequence of both trends, investigated separately; it could also provide answers to the three problem areas mentioned above: in a knowledge-trading society, SME's will discover their strength, namely rapid development of new knowledge in very specific areas, and deal with their weakness, namely lack of knowledge in many other areas, by identifying and buying this knowledge, or by building a new, task-specific, temporary collaboration with a knowledge provider for this specific area. Hence the organizational fragmentation process is to some extent balanced, and KM becomes an inter-organizational topic.

First steps have been made

Of course, there exist already first steps towards the technical and economic goals of Knowledge Trading. Some of them may deal with some facets of the overall scenario, maybe unaware of the bigger picture behind, like researchers building Expert Finder systems: They are mainly thinking about technical solutions for the matchmaking between information needs and expert competency profiles, but completely neglect other areas of knowledge, or all the business-related topics around (market mechanisms, revenue models, etc.). Some parties discovered the good marketing value of the word "knowledge", figured out that there is a whole industry possible, and hurried up to install Internet-based portals and market mechanisms to bring together knowledge sellers and knowledge seekers (cp. [Kafentzis et al., 2002]). However, usually they did:

Existing solutions have serious limitations

- neither take into account the fact that knowledge is not just a book which can be described and retrieved with a simple keyword retrieval, but has manifold complex context and content features which determine its applicability and usefulness in a given situation;
- nor take into account that the real power of electronic marketplaces lies not in copying ways of working known from traditional business (like book selling with a catalog and a simple, sequential seller-intermediary-buyer relationship), but in exploiting the strength of manifold synchronous and asynchronous communication and community-building means, which is of utmost importance when dealing which such a sensible good as knowledge;
- nor take into account that setting up a Web-portal is far from designing sustainable business – which means thinking about customer relationship, advanced revenue models, appropriate pricing mechanisms for different kinds of knowledge and situations, etc.

Besides these limitations with respect to the holistic approach, there are also shortcomings regarding simple metadata aspects. As [Inkass, 2002] shows:

Deficiencies of contemporary online knowledge marketplaces

1. Representation of *knowledge content* is – though *some* of the examined contemporary marketplaces employ interesting metadata sets for their knowledge products – usually weak. Usually, a knowledge product is classified to one or more topics of a (more or less elaborated) hierarchy of subjects. Potential usage context (which may be different from a pure content description in some cases) is very seldom described.
2. Many other aspects (like evaluation of knowledge quality, community aspects, feedback mechanisms, etc.) are either not supported at all, or there is only an implementation of some functionalities which uses implicit data structures not generally known or accepted.
3. None of these marketplace solutions takes into account that in the future there might occur the situation that many knowledge marketplaces exist in the world, such that a need arises for knowledge object descriptions to be exchanged between different marketplaces, or to be integrated from different marketplaces. Hence the idea of an information object as self-contained that it can be shipped autonomously is not yet tackled up to now.

All marketplaces use – of course – quite different metadata sets (or, Information Ontologies), though there is some overlap. Hence the matter of a reusable, standardizable part is still open. Even existing metadata standards or e-Commerce ontologies seem not be used or integrated.

The INKASS approach.

Consequently, the aim of the INKASS project was to develop a total solution for online Knowledge Trading that combines software elements and Business Engineering parts, in particular, consisting of:

Elements of the INKASS solution

- A managed repository of **knowledge products** providing **matchmaking facilities** between the knowledge requirements of buyers and the knowledge products provided by sellers.
- A **business and community infrastructure** to support members participating in knowledge exchange.
- An e-Commerce platform supporting **business models and pricing schemes** for knowledge products.

The role of the information ontology in INKASS

At the core of the "managed repository" – which is implemented as a Case-Based Retrieval (CBR) software – stands a catalogue of knowledge product descriptions which instantiate a metadata schema that is nothing else than an **Information Ontology** as introduced in Section 3.3. This shall be designed to act as a reference model for future Knowledge Trading projects. Because we left the topic a bit vague in Section 3.3, we will here elaborate a bit more on the INKASS Information Ontology.

Its purpose is to provide a declarative specification of the knowledge representation schema used describing knowledge products and the related background knowledge. This shall be the basis for more content-type specific characterizations of knowledge products that allow better search and retrieval; it shall also be the basis for powerful new services (e.g. in the areas of collaborative filtering, or elaborated versioning and evaluation mechanisms); and it shall allow to transport easier an encapsulated Knowledge Object description from one trading platform to the other because it is self-contained to a great extent.

An ideal Information Ontology forKnowledge Trading

Hence, a full-fledged Information Ontology in the "ideal knowledge trading system" comprises:

- A specification of all *attributes* an Information Object[121] for trading knowledge may possess.
- The *value ranges,* and – if necessary – supplementing related ontologies – for defining the ranges of attributes used.
- A specification of all *links and relationships* that may exist between information objects (indicating, e.g., that some knowledge object could provide prior knowledge useful for understanding and applying some other knowledge object).
- The specification of – if required – *aggregated knowledge objects,* represented by aggregated information objects, which deliver some complex piece of knowledge or service by an appropriate combination of several simpler objects (e.g., a series of training measures used for a complex qualification and certification process).
- All other *supporting data structures required,* e.g., for representing contracts or transactions which are required for managing a whole transaction through all its phases before, during, and after selling a knowledge products.
- Ontologies may contain additional supporting information which is exploited by the marketplace for some purpose, like the *similarity between concepts* which is required for assessing similarity of demand and offer representations in a case-based retrieval approach like ours.

Research methodology for designing the INKASS Information Ontology

In INKASS we followed a combined bottom-up / top-down approach to define a comprehensive information ontology for knowledge trading. Bottom-up means concretely that we analyzed the specific requirements of three real-world case studies to be implemented in the project, as well as the metadata schemas found in the existing marketplaces (Inkass, 2002). Top-down means that we analysed both

[121] For the sake of compliance with the INKASS project language, and in order to make life a bit easier, let us call within the context of this section a Knowledge Object Description or a Knowledge Item Description (KID), also an "**Information Object**".

what is provided in an "ideal" knowledge trading scenario and can be derived from our overall trading framework, and what metadata are foreseen in the Dublin Core Digital Library standard, the IEEE Learning Object Metadata standard, and two earlier industrial projects done by the INKASS partners. In detail, for designing the INKASS Information (and related) Ontology(ies) we "compiled" the following input:

Input for the INKASS Information Ontology

- The current state of practice and the acquired requirements of the three INKASS pilot environments at :
  - TWI – selling very specific technology documents, training measures, specifically configured knowledge packs, or consulting services in the area of welding and joining technology to subscribed members
  - Planet Ernst & Young – selling consulting projects to long-term customers
  - ACCI – finding and configuring specific information packs (containing fact books, experience reports, links to relevant events and trade missions etc.) about trading conditions and similar economic information for companies interested in an engagement in aforeign country
- Prior research and customer projects done at DFKI and Empolis.
- The state of the art in the scientific literature, in particular the Dublin Core initiative and the IEEE Learning Object Metadata standard (LOM), as well as some specific approaches for special problems, like IPR representation or contract representation.
- As a further input, we used WordNet (Miller, 1990) which helped us to group and structure certain aspects of content and context descriptions.

Top-level of INKASS information ontology.

Figure 77 below gives an overview of the INKASS information ontology metadata facets.

The vision behind this faceted description is: If all the facets are sufficiently described, it should be possible to assess the content and potential usage and value of a knowledge object comprehensively, to support all processes, transactions and modifications during the lifetime of an Information Object, and to ship such an object self-contained; i.e., transferring it with its complete creation and modification history from one marketplace to another one, (i) without loosing information, (ii) without getting into legal or business problems because of changed contextual factors on another platform, etc.

Of course, this is a far-reaching vision. Nevrtheless, we aimed at a most comprehensive approach which could later be refined and tailored for specific projects. Definitely, the facets described, in particular the details of content and context representation, are a *superset* of what will presumably be used in each specific application case

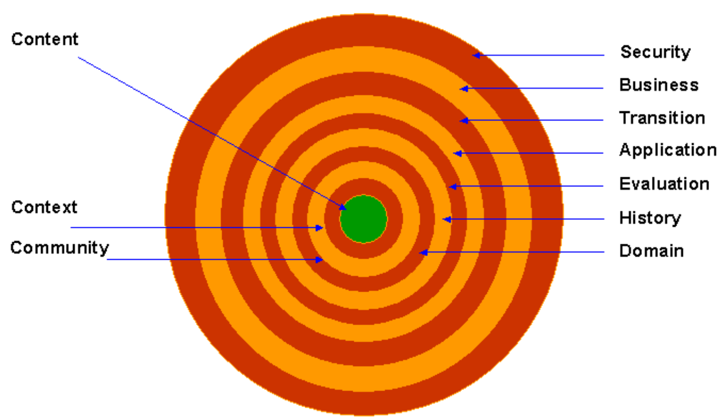


Figure 77: Top-level Structure of INKASS Information Ontology

All description facets are described in some detail in [Inkass, 2003] and partially in [Abecker et al., 2003b]. Here is a short overview over all facets:[122]

- The ***content facet*** shall describe the core content of an information object, i.e. both what it is about (e.g., "this is a textbook about operating systems") and how it is physically manifested (e.g., "the book has 342 pages").

[122] The top-level structure was proposed by [Maas et al., 2003]. The author was mainly further extending, defining and implementing several facets.

- The ***context facet*** shall describe under which circumstances a knowledge product may be used and applied in a customer organization. For instance, we could know that some lesson learned should be useful in all marketing processes of car manufacturing companies.

  > It may be the case that only one of these two central IO description dimensions will be used in a concrete example (e.g., Digital Libraries typically talk only about content, not about context, whereas lessons learned (LL) systems may talk only about the context where some LL could add value), but we discuss both dimensions and feel that it opens promising chances to consider both.

- The ***community facet*** shall address the whole community of agents interacting with an Information Object (IO) representing a knowledge product, i.e. the knowledge providers, disseminators, and users with their roles, rights, and responsibilities with respect to a certain IO. Hence this facet is the interface to all business processes related with knowledge trading. The community facet could define, e.g., that a buyer of some teaching software has the right to use and personalize it, and the right to send bug reports to the programmer (author), whereas the author may have the obligation to inform all buyers about new releases or bug fixes.

- The ***domain facet*** shall ensure that all content-specific statements about an IO are understandable and interpretable even if one transfers the IO from one trading platform to another. Hence it contains the background ontologies or domain vocabularies that define the logical space where an IO and its description facets is situated in.

- The ***history facet*** shall document creation, modification, and change history of an IO, which might be interesting for manifold purposes, e.g. to assess its quality (e.g., think about changes as answers to bug reports or evolving environment or topics) or actuality.

- The ***evaluation facet*** shall contain information suitable to assess the quality of the knowledge represented by an IO. Basically, such information may comprise direct measures describing intrinsic features of an IO (e.g., one may measure the redundancy freeness of a text, the absence of inconsistencies in a formal knowledge base, or the compliance with modeling standards and guidelines for a data model) or its creation process (e.g., it might have been created in an ISO

9000 compliant procedure), or it may contain customer feedback of qualitative (comments of happy users) or quantitative (e.g., a five-star-rating like in Amazon) nature.

- The ***method facet***, or ***application facet*** shall inform about technical provisions required to apply some knowledge described by an IO. For example, in order to use a given PowerPoint presentation, you may need a Laptop with appropriate version of the program and operating system, as well as sufficient memory.
- The ***transition facet*** shall describe how the application of some knowledge may affect and change the application environment. A typical example comes from the e-Learning area: in order to apply a learning object (LO) (e.g., consume some lesson) you are supposed to have some prior knowledge level, and appropriately applying the LO will change your level of expertise, e.g. such that you may subscribe to an examination.
- The ***business facet*** shall be used to store all data and information used to establish the trading functionalities of the marketplace, in particular pricing information.
- The ***legal aspects*** shall comprise everything related to legal aspects of knowledge trading transactions, i.e. in particular all IPR (Intellectual Property Rights) issues affected.
- The ***security facet***, finally, shall represent all information required to ensure that the whole transaction on the web is secure, e.g., with respect to payment and knowledge transfer.

In the context of this thesis, the following status of the INKASS Information Ontology work can be reported:

- ➢ The security and the legal facet were not yet investigated thoroughly enough.
- ➢ For the community, the history, the evaluation, the method, the transition, and the business facet, first suggestions have been made in [Inkass, 2003]. These suggestions were based on straightforward technical ideas and on solutions for similar problems in e-Learning and e-Commerce. Mainly, we proposed data structures for realizig declarative, ontology-based, self-contained specifications of the

respective aspects. However, going into any detail would go beyond the scope of this thesis.

➢ The remaining two facets which are interesting in the context of this thesis, because they illustrate in much detail the topics which were only indicated in Chapter 3, are the content and the context facet. Hence we will describe them more detailed below.

Overall modelling rationale.

Before we discuss some details of our Information Ontology, let us briefly describe the overall modelling approach.

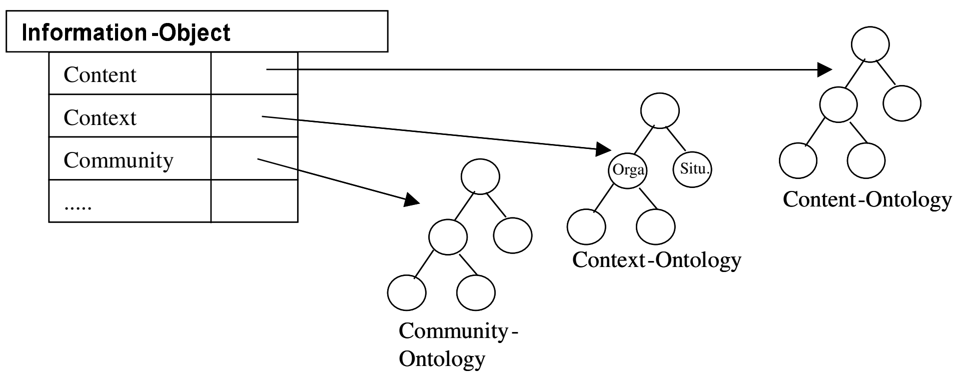


Figure 78: Information Objects and Facet Ontologies

The information ontology defines a top-level structure further refined in facet sub-ontologies

Figure 78 illustrates the first design decision for the INKASS Information Ontology: All facets used for describing an IO take their structure and values from specific sub-ontologies. For instance, a content ontology may provide the domain specific means for describing a document content. A context ontology might define that organizational (in which department to apply some piece of knowledge) and situational aspects (in which business process and to which purpose to apply some piece of knowledge) may be relevant to describe a potential usage context for an IO. In this respect, we are fully aligned with the approach presented earlier in Subsection 3.3.

Since our aim is to impose as much structure as possible on our models in order to make explicit as many design decisions as possible; further, to facilitate the exchange of specific parts for a given application; the technical means for these purposes is partitioning the IO description into specific partial description objects

Figure 79). These description objects group the most generic parts of sub-ontologies.

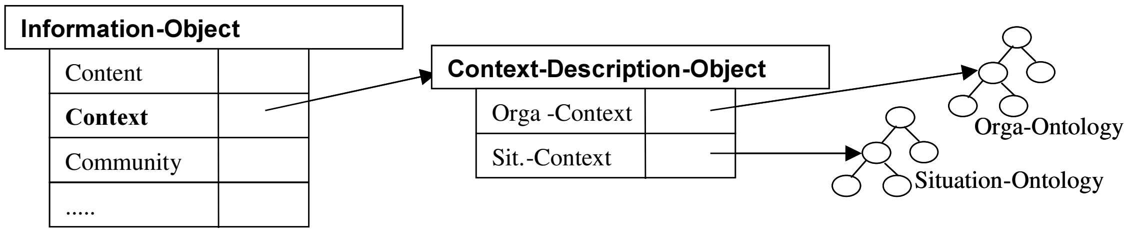
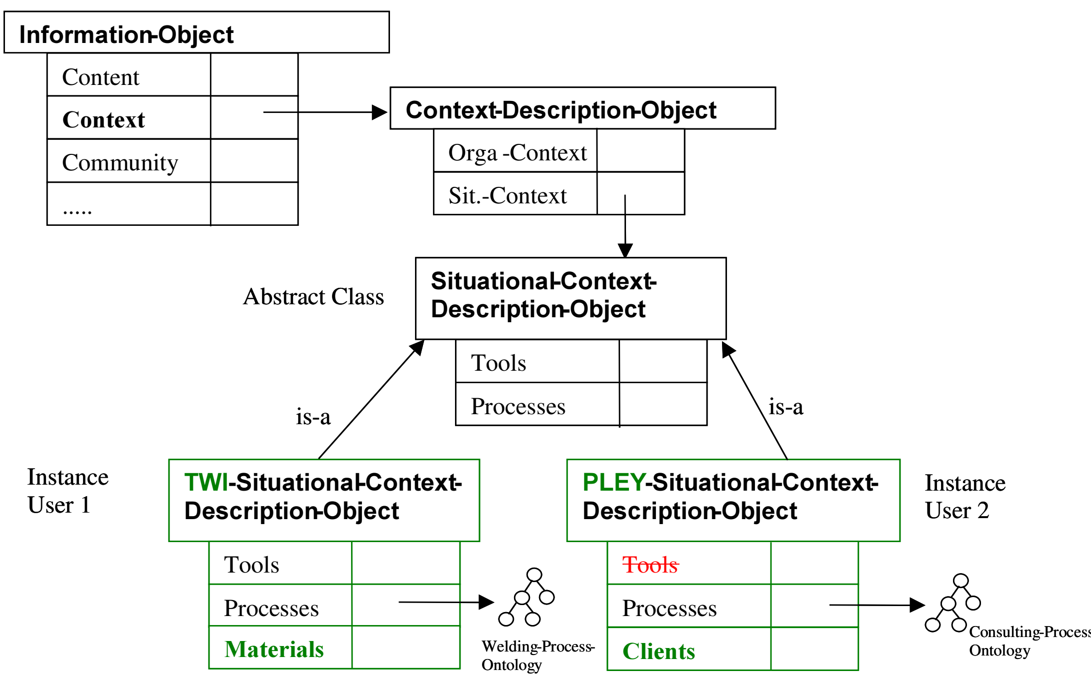


Figure 79: Partial Description Objects Give Structure to the IO

Now a concrete instance for an application case does not directly instantiate the classes defined so far. Instead, we introduce an abstract intermediate class which provides some standard attributes for describing a given sub-facet (like situational context). This abstract intermediate class will be overridden by application-specific instances which may remove certain attributes or add application-specific ones, as shown in Figure 80 (see below for a concrete example).[123]



[123] Please note that this is an is-a relationship which allows overriding. Clearly, when one defines a comprehensive reference ontology, it will happen that in a concrete application, parts must be cut off (as in the figure, the "Tools"), parts must be specifically instantiated and refined, and parts should be reusable.

Figure 80: Case-Specific Description Objects

Detailed Description of Information Object Facets.

Now let us come to the two facets of the INKASS Information Ontology which we want to examine a bit deeper in this thesis, the content and the context facet. For both, we first introduce the general structure, and then illustrate the above-mentioned mechanism for case- and domain-specific refinement.

Facet 1: Content.

General definition of the content facet

The content facet is the core of an information object. It contains the content of an IO – if electronically available – plus metadata describing the kind of content, what it is about, and how it is physically manifested. Attributes and values for these descriptions are defined by corresponding content ontologies.

General structure of content facet.

Figure 81 shows the top-level structure for the content facet of an IO. [124]

[124] The format of the figures is the output of the Protégé Ontology Modelling Graph Visualization tool. Boxes represent concepts or instances. Below the name, we find a number of relations. Relations are characterized by their name and their codomain, if they take a value from a basic datatype (such as the Content_URL which takes a string as a value). If the codomain is a defined concept, the middle column specifies whether the associated value via this relationship may be an instance of this concept, a set of instances, or a subconcept of this concept. If there are relationships which have defined concepts as codomain, these concepts are associated in the picture via a named link (named with the relationship name).

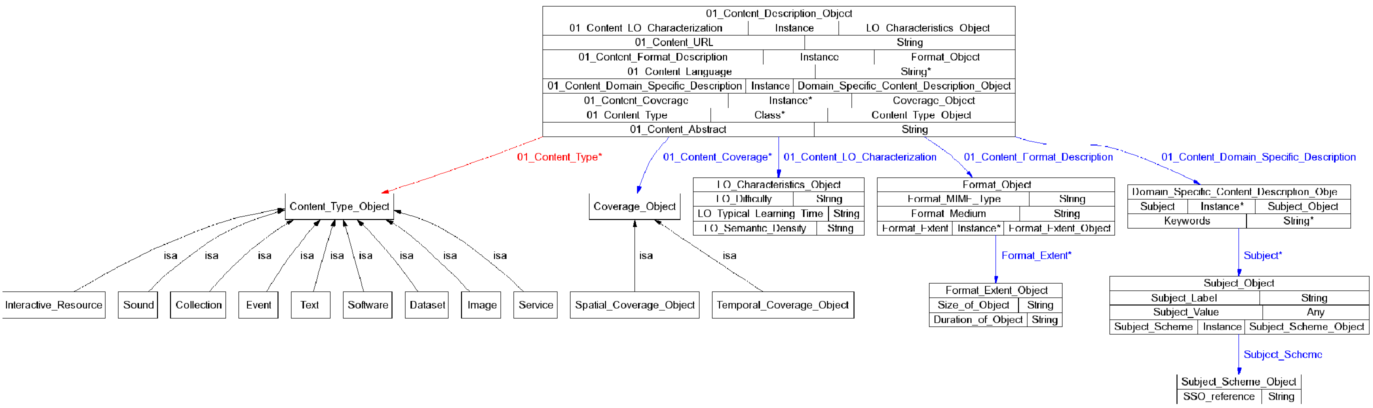


Figure 81: : Top-level Structure of Content Description

As default attributes we provided the following ones which are directly taken from the Dublin Core (DC), or the Learning Object Metadata (LOM) standard, respectively, such that we don't explain them further in this document (we did not model in all details these attributes, since they were not of great importance in the INKASS pilot applications):

- URL: specifies where to find the core content
- Abstract: a short textual description of this content
- Content language, to be encoded according to ISO 639-2 "Codes for the representation of names of languages" and RFC1766 "Tags for the identification of languages"
- Content format, according to DC: MIME and IMT encoding schemes.
- Content coverage, addressing time and spatial coverage, according to DC: DCMIPoint, ISO 3166 "Country codes", DCMIPeriods and W3C-DTF
- A subset of the Learning Object content characterization, according to LOM.
- A domain-specific content description, similar to the DC approach, with freely chosen keywords or subjects encoded according to LCSH, MeSH, DDC, LCC, UDC.
- The content type, as one of the nine content types proposed by DC

DCMIType.

Now, in addition to these generic structures, there may come application-specific extensions and adaptations. With respect to content characterization, we have the following changes in our cases:

Case-specific adaptations of content facet.

Besides the fact that many attributes will not be used in our three cases, the most important adaptation is that, for all three cases, we have specific content types characteristic for each application case. Table 30 lists the different kinds of Knowledge Objects (i.e. content types in the terminology of the Information Ontology) which have been identified in the TWI application about technology consulting.

Case Studies:	Description of a concrete problem and how it is solved
Relevance Packages:	List of advantages and facilities a technology could provide in a certain context
Literature Summaries:	Summaries of the relevant literature
Structured FAQ Collection:	Concise answers to frequently asked questions
Knowledge Summaries:	Brief information on the most popular processes, technologies and materials. Essential Knowledge, risks and benefits relevant to a technical area.
Suppliers Data:	Relevant data about the major suppliers
Standards & Directives Package :	Collection of different Standards and Directives which are relevant in the specific context
Training needs assessment:	Online multimedia training courses
Best Practice Guide:	In-depth guide to technologies and processes which allows a broad comparative look across a field
Research Reports:	Different reports and articles about the relevant results of research
Commentaries:	Comments made by an expert at a specific document
Recommendations:	Collection of documents which are recommended by an expert

Table 30: Knowledge Object Types in the TWI Application

In the Planet-EY application in the area of management and IT consulting, the INKASS tools shall be used in-house in order to react fast and precise when a customer request for quote or telephone request comes, and one has to find out how to approach a given problem. Table 31 presents the kinds of knowledge objects identified.

Methodologies:	Different consulting methodologies, which describe the possible actions the customer can undertake to examine his or her problem.
Assessment or analysis tools (consulting tools):	Software or document-based tools assisting service delivery throughout the project life cycle.
Training material:	Traditional or electronic learning possibilities.
Expert profiles:	CVs of the consultants, especially a description of the projects they worked in.
Links to related documents and sources (internal/external to the organisation):	Internal and external links to documents, papers, books, external experts, websites etc.
Case studies:	Documented examples of service delivery in real cases.

Table 31: Knowledge Object Types in the Planet-EY Application

Finally, Table 32 deals with the ACCI scenario which is about a company that wants to find out which Knowledge Objects could be useful when they plan to start a new business in a certain foreign country.

Executive Summaries:	The most crucial points of extensive reports
FAQs:	Concise answers to frequently asked questions organized around areas of interest (e.g. company formation or tax)
General Reports:	Booklets providing information on the political and economic status of a country
Investment Statutory Framework:	The existing framework that rules foreign investments in the destination country.
Bilateral Trading Agreements:	Existing agreements between the governments of the two countries regarding investments, commerce and industry.
Trade Missions Reports:	The outcome of the various trade missions organized by ACCI in foreign countries.
Contact points:	A list of essential contact points (chambers, embassies, ministries etc) in the foreign country

	ministries etc) in the foreign country.
Internal Research:	Results from research among the members of ACCI that have done business in a country
Case Studies:	Description of problems that the enterpreneurs have faced and how they solved them (Oral and Informal).
Activities Survey:	I Information on missions to be organised and important events (fairs, sampling visits etc).
Commentaries:	Comments made by an expert at a specific question.
BusinessNet:	Relevant data about traders.
Contracts:	Model Forms for Agreements/Contracts for various types of transactions

Table 32: Kinds of Knowledge Objects in the ACCI Case

Now, all the listed kinds of Knowledge Objects should go into the case-specific adaptations for the respective cases, as specific values for the "content type" attribute. For the sake of space, we don't show all extensions here (they are documented in [Inkass, 2003]), but Figure 81 gives at least some idea of the top-level structure of the content ontology, extended for the three case studies.

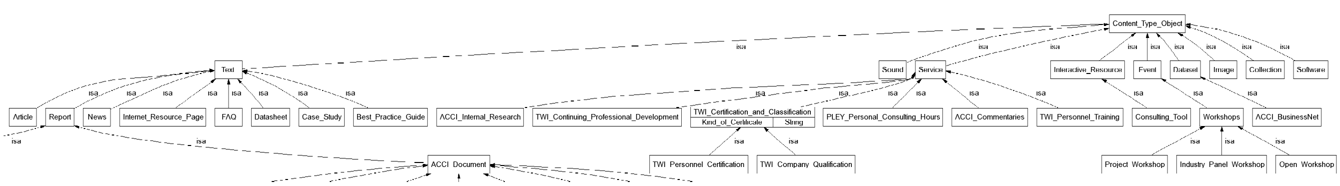


Figure 82: Some Case-Specific Adaptations to the Content Facet

Facet 2: Context.

General definition of the context facet

The context facet describes the application context in which a particular information object can be used. Context information is described by and linked to context ontologies.

Simple Example

A lecture, for instance, can be used in a teaching context at a University for MBA courses. If an information object is context-independent it is described by ‚any' context.

Clarification remarks

First we have to note that content and context may be different even if they are not necessarily both existing and different in concrete application examples. Especially for consulting projects or lessons learned from consulting projects, it might be difficult to make a distinction between content of a project and potential application context. We can argue abstract that both may differ if we think about Storytelling (Denning, 2000) as a widespread Knowledge Management technique. There, content and application context are more or less by definition different. As an even simpler example, take fairy tales for children. There, the content of a story may be about dragons, knights, or witches, but the application context may be to teach children about courage and daringness, about love and loyalty, or about revenge and pardon.

Second, we note that we consider the fact of having both content and application context at our disposal, as a really important feature which shows the direction for future Knowledge Management and e-Commerce scenarios. The reason for this is that often a user *does not know* what the content of a knowledge product useful for him could look like. He does not know the answer, but he knows the problem! This means, what we need is describing problem situations (i.e. potential application context) and linking them to possible solutions.

General structure of context facet.

Figure 83 shows the general structure of the description of potential usage context. It is composed from two parts:

- the static context, i.e. the *organizational context* in which a knowledge product may be applied; and
- the dynamic part, i.e. the concrete *dynamic situation* in which a knowledge product may add value.

Organizational context

The **organizational context** shall describe as comprehensively as possible an intended consumer of a given knowledge product. Currently we foresee the following attributes to realize such a comprehensive description:

the <u>intended User_Organization</u>, if some knowledge product is produced exclusively for specific customers, or if its applicability depends on certain customer company characteristics, like the size, the location, or the legal form of a company;

the intended User_Department within this organization, because a given knowledge product may only make sense to be used by the marketing department or the production planning;

the Organizational_Role(s) which may apply a knowledge object successfully (because they have the competencies, rights, or responsibilities to do so, or because a knowledge product – like a lesson learned or a best practice – affects in particular their specific job); and

the Age and professional Experience of the people in these Organization_Roles, because there might be preconditions which must hold to employ a knowledge product effectively (for instance, such conditions frequently exist in the TWI case)

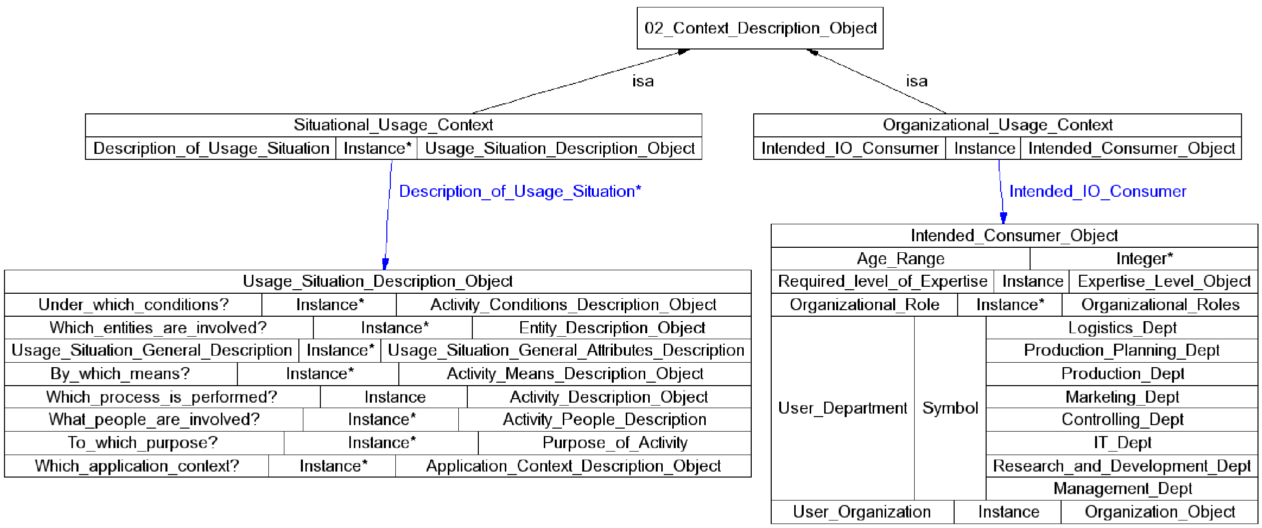


Figure 83: Top-level Structure of Context Description

On the other hand, the **dynamic, situational context** is constituted by following attributes trying to describe as detailed as possible what activity shall be executed in which manner. For this description, we oriented ourselves on the classical "W-Questions", Who, What, When, … . Though this will be clarified better below, when coming to the case-specific instantiations, we shortly summarize these attributes:

We describe which process (e.g. a certain production process) is performed, manipulating what entities (as input, output, or auxiliary products), under which conditions (e.g., obeying to specific regulations with respect to health or environment), and to which purpose, by which people, through which means, and in which general application context (e.g., the industry sector).

These dimensions can be further decomposed and will be instantiated / adapted for specific case. We show the general principle by means of the TWI case which is the most developed case in this respect.

Case-specific adaptations of context facet.

Figure 83 shows the general approach how we achieve as much generic structure as possible, yet being able to add application-specific extensions and refinements wherever required: The value range for all complex attributes (namely, User_Organization, Organizational_Role, and Required_level_of_expertise as general attributes for the Organizational_Usage_Context) is specified by an abstract class for objects to describe the respective sub-facets. These abstract classes may provide a default model using the most generic or widespread attributes, but will usually be specialized to case-specific classes which may add or remove attributes, or change value ranges or default values. In Figure 84 we see an excerpt of the implemented model which contains some case-specific extensions from the LOM e-Learning area, the HAL application (an industrial project brought into the INKASS research as an input by Empolis), and the TWI case.

To mention just one example for the TWI-specific extensions: If we consider the sub-facet Organization_Object of the intended organizational usage context of some knowledge product, the generic range of this attribute is a general class for organizations, in the most simple way represented, e.g., by their name and their address. Now, we can define a TWI-specific specialization which inherits these attributes, and adds two new ones:

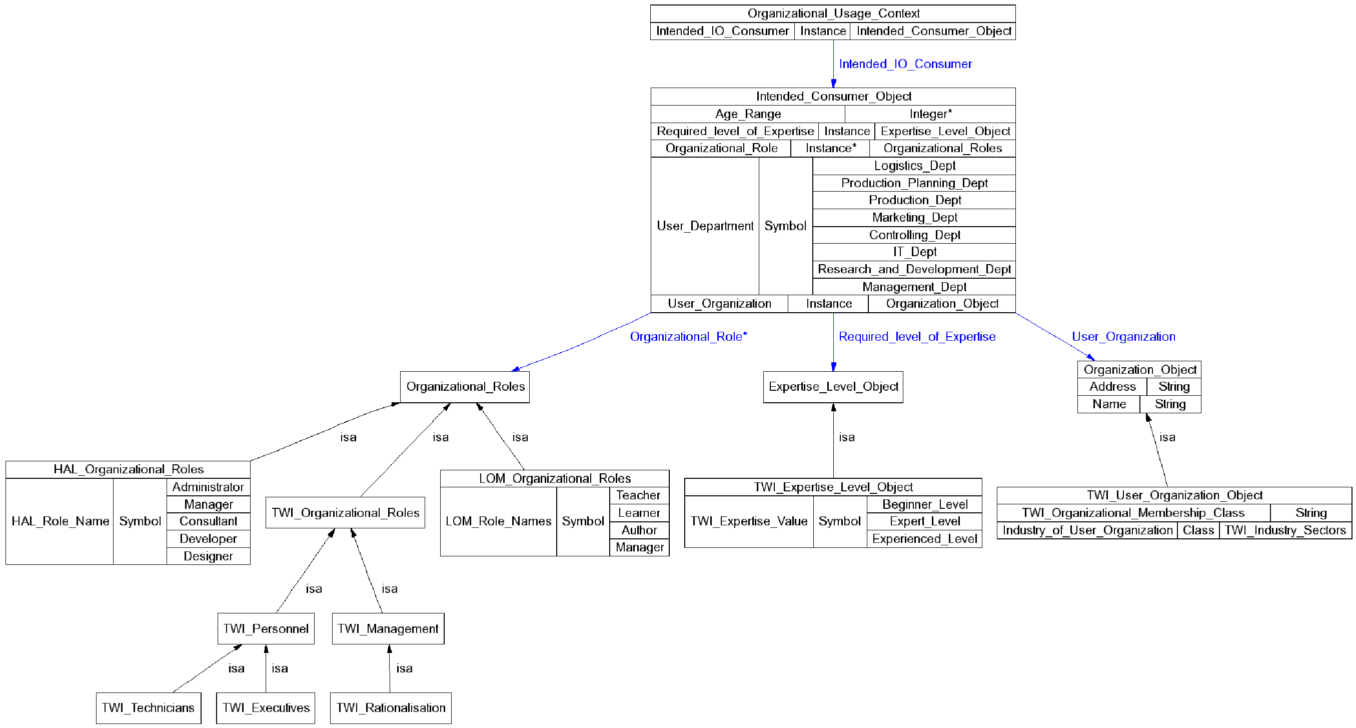


Figure 84: Generic Structure for Organizational Usage Context, plus Case-Specific Extensions

- The TWI_Organizational_Membership_Class[125] describes the specific kind of organizational membership which may affect availability and access to certain knowledge products (in the most simple case: companies which are not members of TWI don't have access to many documents and services; similar mechanisms exist in ACCI).
- Further, it is highly relevant for TWI in which Industry_Sector a given user organization is active. In Figure 85 we indicate the industry sectors used by TWI to structure their view on customers, very similar considerations hold true for both Planet-EY and ACCI. Figure 86, for example, shows an exemplary part of the Industry_Sector part of the Planet-EY domain ontology. As a side remark it should be noted that in both cases there were already established models in place which could be reused and integrated in the INKASS ontology. For transferring the INKASS solution,

[125] A company may become a member of TWI (there are different classes of membership, with different fees) then having easier and cheaper access to the more sophisticated kinds of TWI knowledge and services..

it would probably make sense to incorporate some standard Industry Sector model from e-Commerce standardization initiatives as a default model to reuse or adapt for a specific new application.

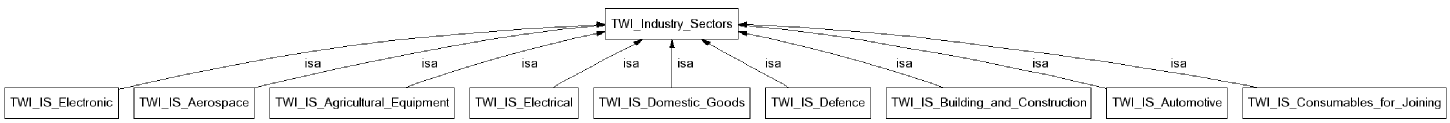


Figure 85: TWI-Specific Industry Sectors

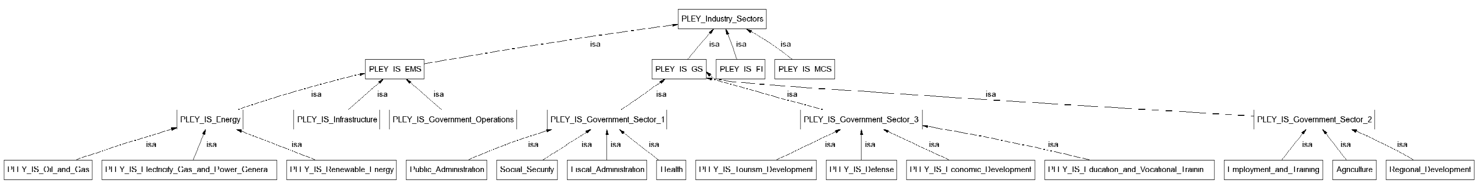


Figure 86: Planet-EY Specific Industry Sectors

Regarding the **situational aspects of the usage context** dimension, similar structures can be identified. Figure 87 shows the top-level structure of a situational context description which introduces for each of the above mentioned facets a class for declaring possible attribute values. These value range classes further decompose some facets, like e.g.:

- The question by which means an activity is performed, may be decomposed into the attributes Activity_Method_Used (e.g., the Balanced Scorecard Method to assess Intellectual Capital), Activity_Tool_Used (e.g., a specific CASE tool for software development), the Activity_Equipment_Used (see below for the TWI example), and the Activity_Technology_Used (ditto).
- The conditions under which an activity is performed may be specialized to the standards the activity enactment complies with (e.g., an ISO9000 project, or a UML-based software design).
- The entities to be dealt with in an activity may be refined into input and output products, described, for instance, also with some properties (see TWI example).

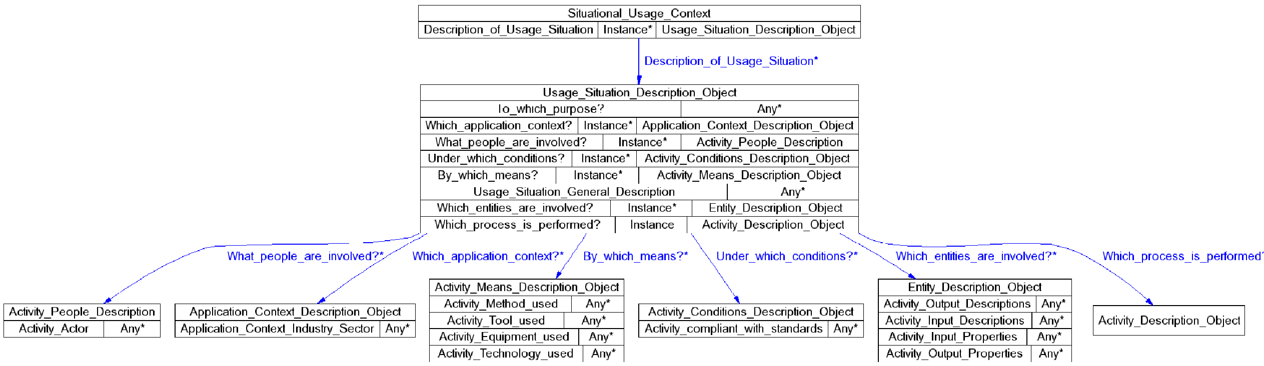


Figure 87: Top-Level Structure of Situational Context

In order to finally come to the concrete attributes to be used in a specific application, we can further specialize the leaf classes in Figure 87 for producing application-specific attribute sets and value ranges. We will explain this again with some examples from the TWI case which is the most elaborated in this respect.

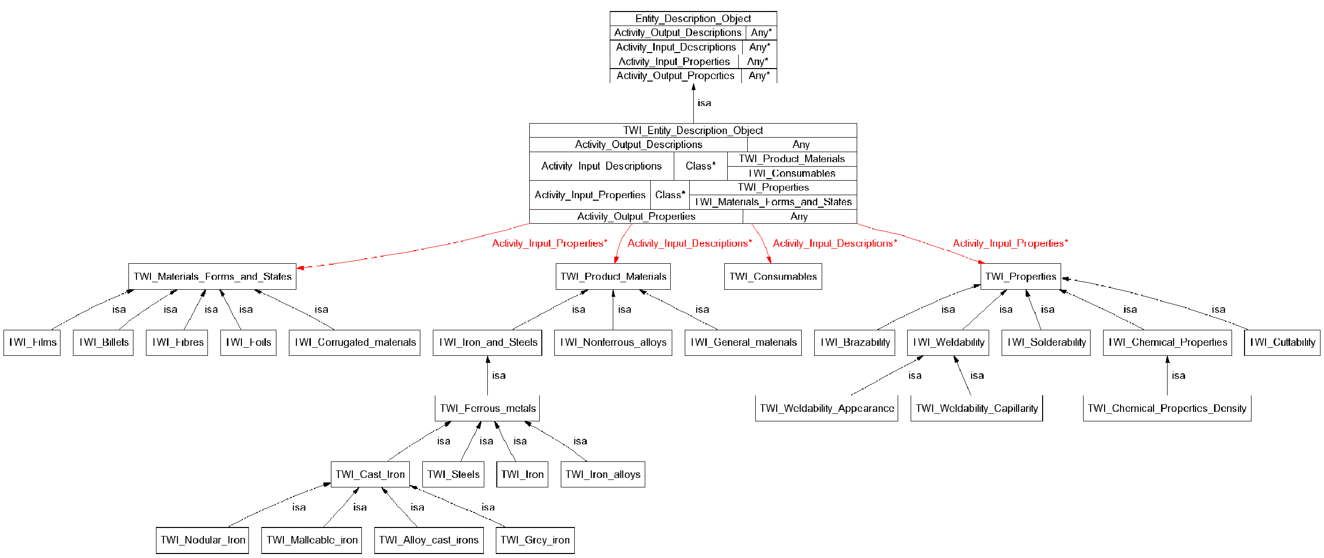


Figure 88: TWI-Specific Classes for Entities Involved in an Activity

Here (Figure 88), we specialize the generic structure of an entity description insofar as we further refine the way we can describe input products for a certain activity. Instead of allowing arbitrary objects as values for the attributes Activity_Input_Descriptions and Activity_Input_Properties, we link into the model very specific parts of the TWI domain ontology here:

1. For the Activity_Input_Descriptions attribute, we allow an object descri-

bing a TWI_Product_Material, or an object describing a TWI_Consumable. Both alternatives take a value from a TWI specific class which is elaborated in the figure to a small extent. In the fully implemented application, the TWI_Product_Materials and TWI_Consumable sub-ontologies are, of course, much bigger.

2. For the Activity_Input_Properties attribute, we also allow values from two domain specific sub-ontologies: TWI_Materials_Forms_and_State describing that we are interested in a concrete application situation, e.g., in specific forms of consumables; and TWI_Consumables – not further refined in the example – listing the different possible types of consumables for a process considered by TWI.

While this dimension ("entities") tackles the question "with what" we do something, there is of course also the "what do we do" question.

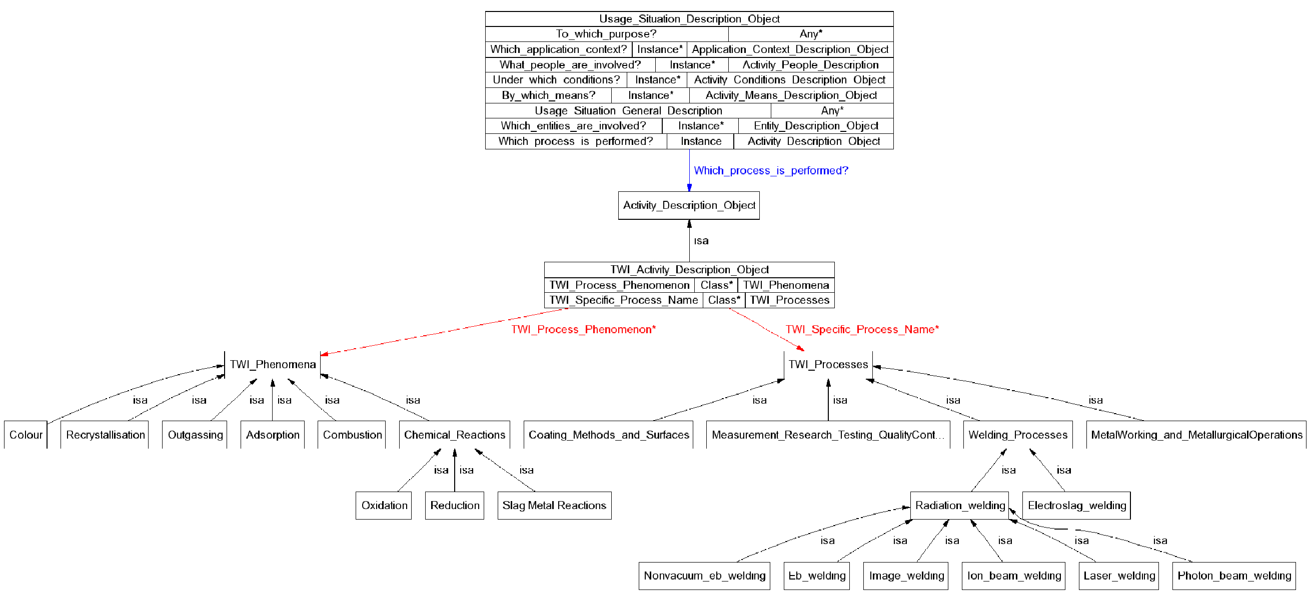


Figure 89: Activity-Description, Specific for the TWI Pilot

Figure 89 illustrates how this question may be answered in the TWI case: an activity is described by the specific technological TWI_Process and the TWI_Process_Phenomena that we are interested in for a specific knowledge product. Here, we have again the link between Information Ontology and TWI-specific domain knowledge (kinds of processes, kinds of phenomena). Of course, also here, the TWI domain ontology is only shown to a small extent. The first-cut implementation in INKASS, as built from existing data structures, expert interviews, and document analysis, contained, e.g., 500 Processes in 9 top-level categories and 60 Process Phenomena in 3 hierarchy levels.

In contrast, for the Planet-EY case, we don't have such a specific attribute as the TWI_Process_Phenomena, but we have also a Planet-EY specific list of customer processes a consulting engagement wants to support (see Figure 90).

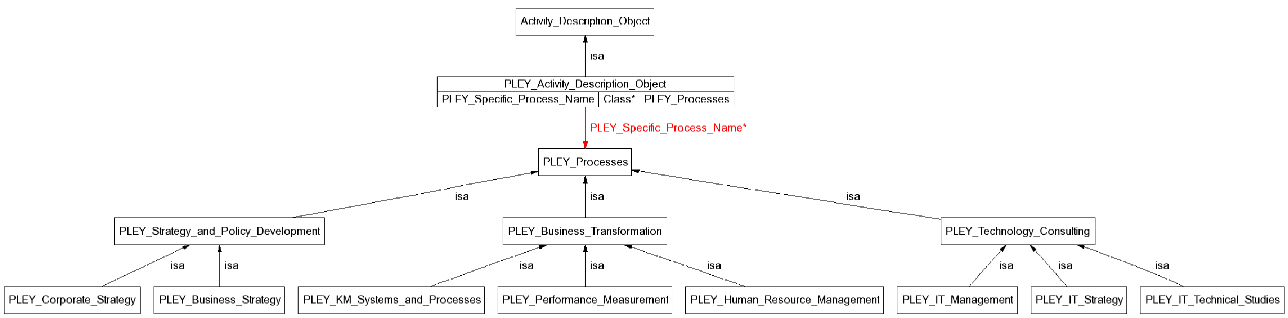


Figure 90: Activity-Description, Specific for Planet-EY Pilot

Another sub-facet of situational context may be the overall application context a specific organizational activity is embedded in. Figure 91 shows the TWI specific refinement which distinguishes between Industry_Sector where the knowledge should be applied – a quite rough classification of the application environment, and the Fields_of_Application, which is a more specific description of the subject matter addressed.

Generally, the distinction between these two sub-facets is not necessarily understandable and acceptable from the ontological point of view for a naïve user, but it represents the way that TWI people and their customers agreed to see the world. Here we see that customer specific adaptations are easily embeddable in the general approach.

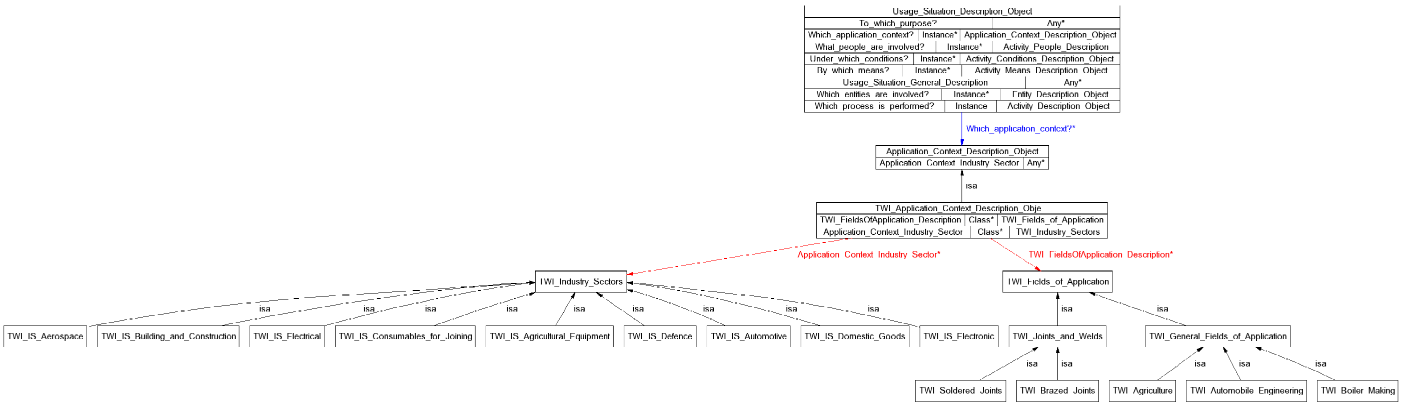


Figure 91: Description of Application Context, Specific for TWI

If we take the Planet-EY case as another example, we can see that there – besides the Industry_Sector of a potential customer organization and besides the PLEY_Fields_of_Application which describe the potential subject areas where a Planet-EY consulting engagement could work about – a new attribute is added, namely a project as a potential context factor. If we represent projects also as cases like all other data in our system, specifying an example case where some experience could be (or, was) beneficial allows to assess the relevance of this experience in a new situation also on the basis of making a comparison of project similarities (see Figure 92).

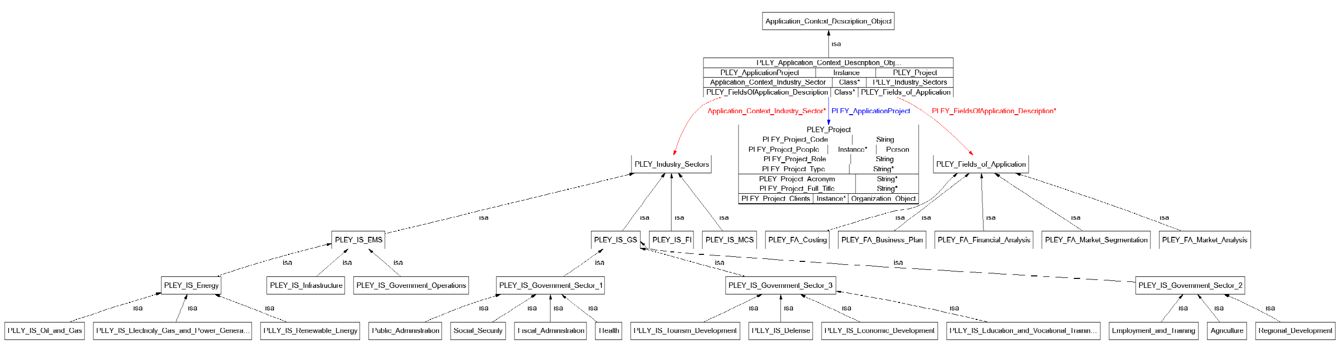


Figure 92: Description of Application Context, Specific for Planet-EY Pilot

The next sub-dimension of situational usage context is the question by which means we want to achieve a goal. In general, we foresee attributes for Methods, Tools, Equipment, and Technology used. Figure 93 shows how this aspect is specifically realized in the TWI case:

- Method and Tool are not that relevant in this case here, and are thus not used. This is indicated by the fact that the two slots inherited from the generic superconcept, Activity_Method_used and Activity_Tool_used still have the generic, unrestricted value range “Any”.

- However, the two other slots, Activity_Equipment and Activity_Technology, are used and their values can be taken from two hierarchies taken from the TWI domain ontology, namely the TWI_Technologies and TWI_Equipment. Again, both are shown in the example only to a small extent. In the first-cut TWI system implementation, the TWI_Technologies hierarchy contained 160 concepts, and the TWI_Equipment hierarchy contained 40 concepts.

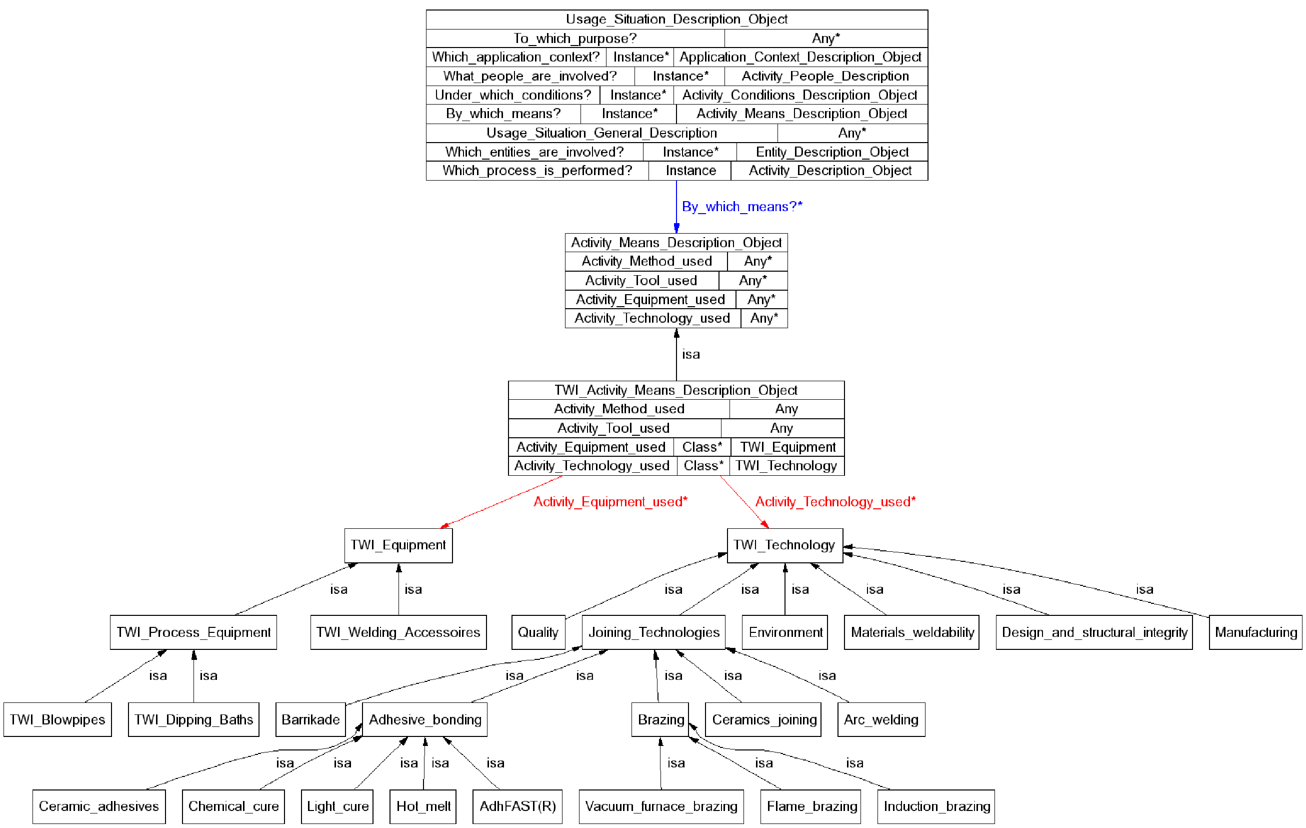


Figure 93: Describing Means for an Activity, Specific for TWI

While the sub-facet described above considers the question with which material and immaterial tools an activity is performed, a further dimension of analysis may describe the conditions under which an activity takes place. Figure 94 illustrates how this sub-facet is used in the TWI case: while the “Standards_to_be_-

Compliant_With" are not used in this case, instead new, TWI-specific ones are added which further specify in which details of an activity one is interested.

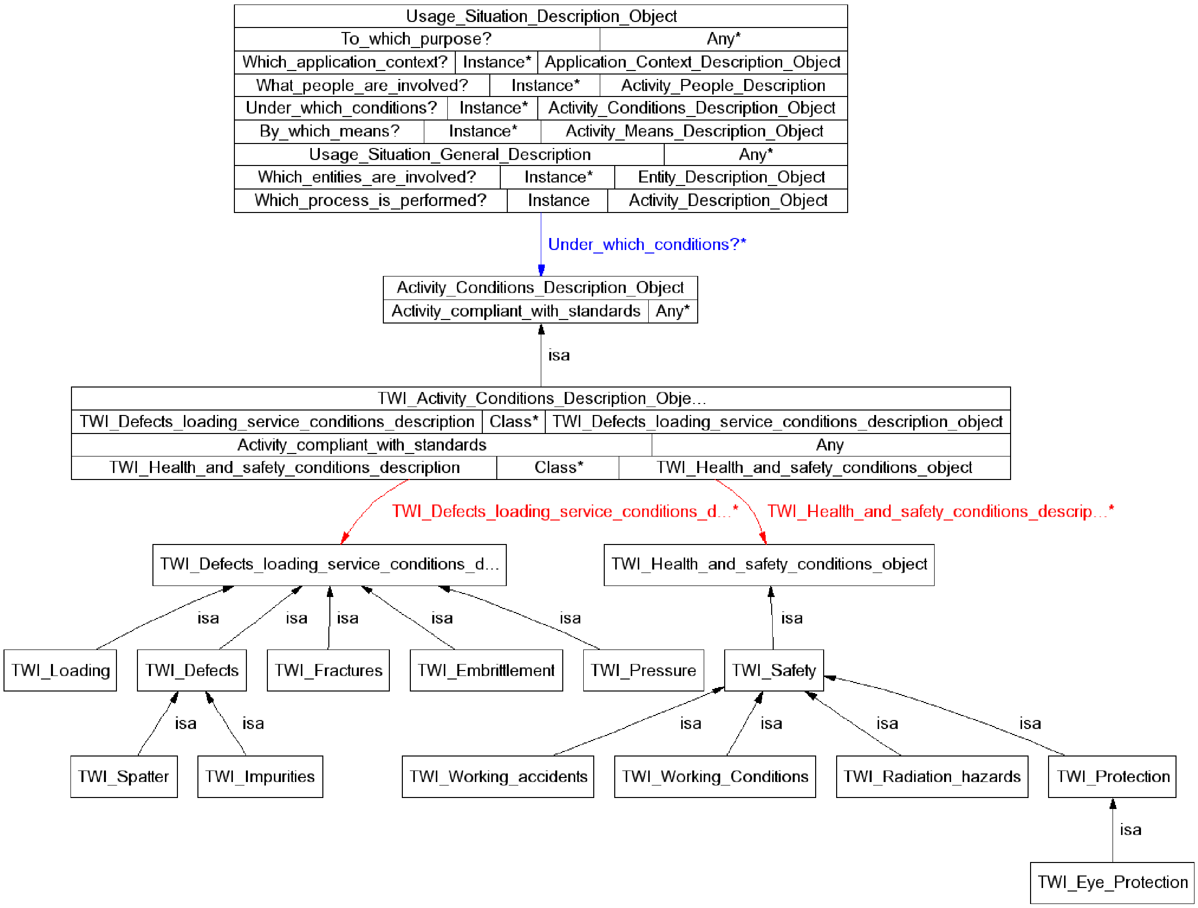


Figure 94: Activity Conditions, Specific for TWI Pilot

As another example take Figure 95. Here we have the activity conditions relevant for knowledge about software products. In this case, we do not add new attributes, but we specialize the range of allowed attribute values to a (sample) set of software standards.

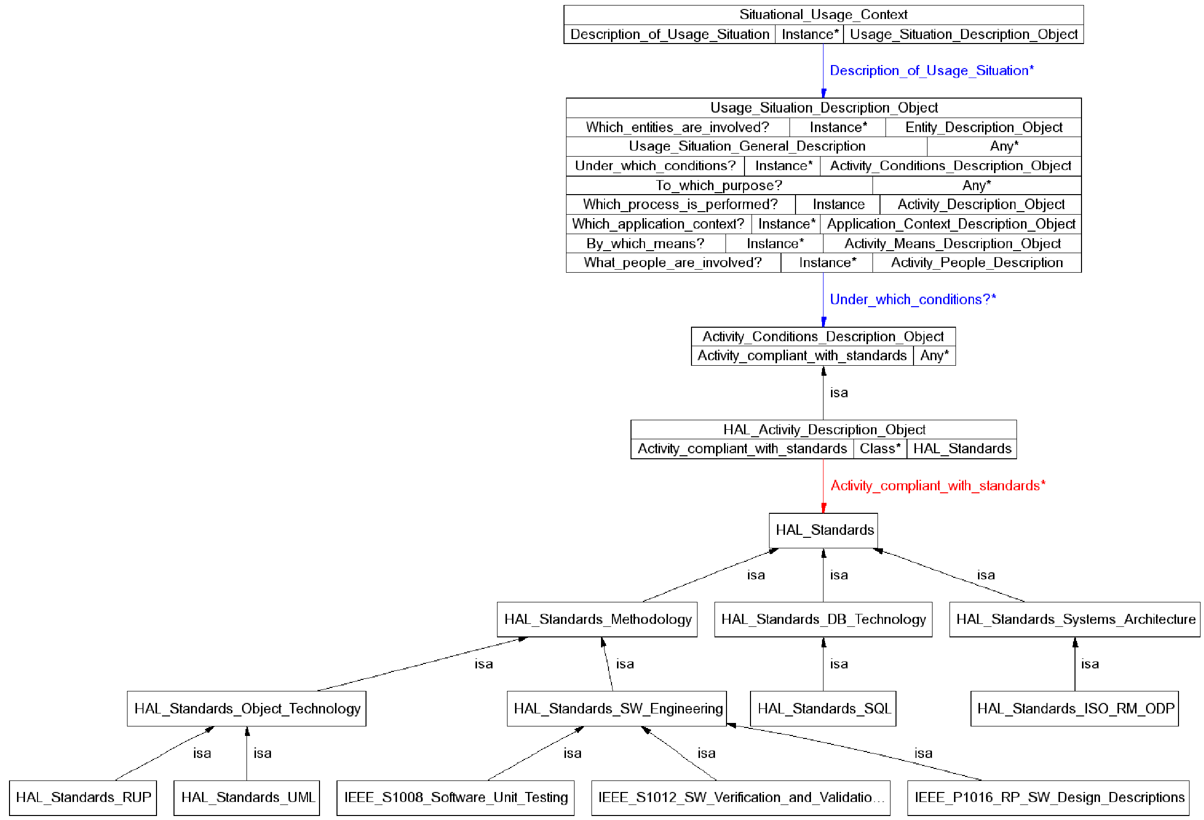


Figure 95: Case-Specific Activity Conditions, for Software Projects

Regarding the purpose of an activity, Figure 96 shows an initial decomposition of purposes that an organizational activity may have.

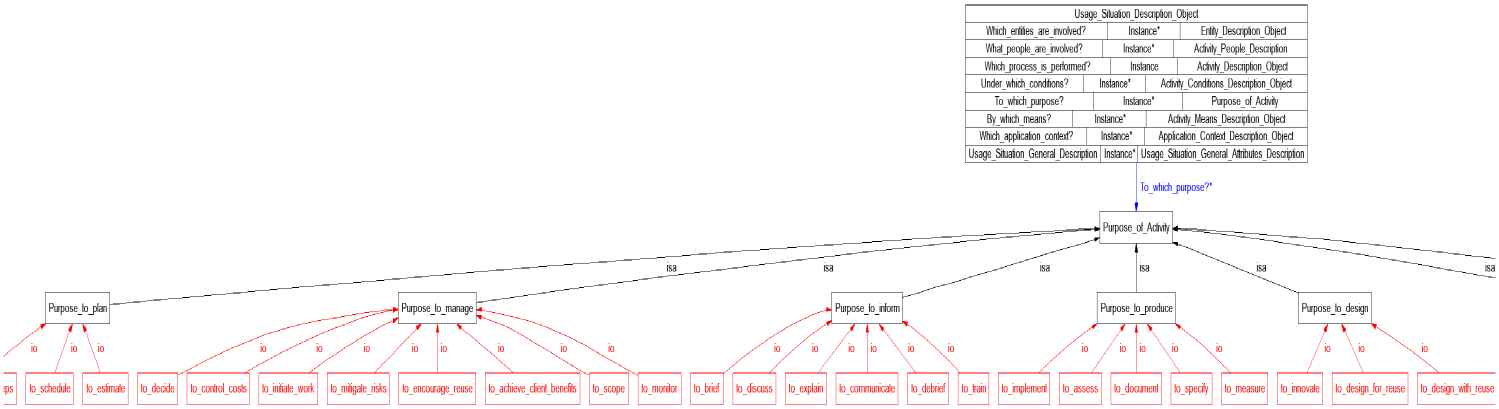


Figure 96: Possible Purposes of an Activity

This should be enough information to get an impression how our reference ontology is designed and how one would work with it. For more information, please refer to the INKASS project documentation.

5.2 Agent-Mediated Knowledge Management[126]

Up to now, all our OMIS considerations were based on the assumption of one, centralized repository system with globally valid ontologies and structures. However, such a *centralized* OMIS approaches may have drawbacks with respect to several aspects:

Local versus global concerns should be traded off

- Knowledge generation and use in an enterprise is *distributed by nature*. Departments, groups and individual experts develop individual, differing views on given subjects. These views are motivated and justified by the particularities of the work, goals, and situation in question. Obtaining a single, globally agreed-upon vocabulary on a level of detail which is sufficient for all participants may become expensive or even outright impossible. Consequently, an OMIS should benefit from balancing both *local expertise* –which might represent knowledge which is not easily shareable on a global level–and *overall views* on a more global level. A strict centralized approach neglects this opportunity.

Central architectures are often bad prepared for change

- Knowledge resides in changing environments. A centralized OMIS may be ill-suited to deal with continuous modifications in the organization: The maintenance costs for its detailed models and ontologies might get too high.

KM should evolve from many "quick wins"

- Furthermore, centralized OMISs assume a strict sequence of design, implementation, and use. In practical projects, a more evolutionary approach seems rather promising: OMIS-like structures evolve in different groups and departments, using appropriate formalizations and conceptualizations. Integrating these elements under a common roof without disturbing their individual value should result in solutions which offer common benefit with reduced efforts – while reaching better acceptance on the individual level.

---

[126] Parts of the motivation of this Section have been published in prior versions in [Elst & Abecker, 2002a; Abecker et al., 2003c].

Inter-organizational KM becomes more important

- Last, but not least, the fragmentation trends in economy with their counter activities in form of closer cross-organizational collaborations (that we mentioned already for motivating Knowledge Trading) and inter-organizational KM, let appear appropriate to think not anymore about *an OMIS,* but rather about *dynamic societies of cooperating OMISs*.

The reality of enterprises' environments thus asks for a *distributed* approach to OMIS realization: Distributed, heterogeneous OMIS cells let local expertise prevail while striving for maximal integrated benefit. Evolutionary growth and scalability on all levels is reached by allowing individual OMIS cells to grow and mature independently, while interaction and communication brings enterprise-wide exchange and understanding. In [Elst & Abecker, 2002a], we give some examles that in such a distributed scenario even different layers of several OMIS installations could benefit from each other (we showed that the Knowledge Broker Layer of one system might want to get input from the KOL, the KDL, or the KBL of another OMIS).

The OMIS scenario asks for intelligent agent technlogy

Now taking into account that applications which are modular, decentralized, changeable, ill-structured, and complex, are typically considered ideal application fields for Intelligent Agent technology, it is nearby to think about agent-based OMIS implementations (cp. [Parunak, 1998], we elaborate a bit more on the applicability of agents in [Elst & Abecker, 2004; Elst et al., 2004b]).

A simple agent definition

Following [Wooldridge & Jennings, 1995], we assume the "weak definition" of software agents with the definitional features *autonomy, social ability, reactive behaviour,* and *proactive behaviour*.

At least, we can make the interesting observation that (partially, already for a long time) in several research areas, all required elements of an OMIS implementation have already been realized with agent technology (see Table 33).

Application Layer	Workflow agents, task and process agents	[Joeris et al., 1997], [Jennings et al., 1996]
	Exploitation of personal work context and context-sensitive information provision with interface assistants	[Budzik et al., 2001; Budzik et al., 2002 ; Bauer & Leake, 2001]
	User profiling agents & personal assistants	[Bauer & Leake, 2002; Müller, 2002]

Knowledge Broker Layer	Knowledge push/pull mechanisms	[Mahé & Rieu, 1998]
	Digital reference and acquisition librarians	[Carbonell, 1996; Stojanovic, 2003]
	Mediators & facilitators, ontologists, knowledge brokers	[Wiederhold & Genesereth, 1997; Fensel, 1997; Andreoli et al., 1995]
	Cooperative information retrieval	[Decker et al., 1995]
Knowledge Object / Knowledge Description Layer	Agents for document analysis + information extraction	[Nakata et al., 1998; Klein & Abecker, 1999; Eliassi-Rad, 2001]

Table 33: Software Agents Realizing OMIS Functionalities

Seeing that almost all individual functionalities have already been realized somewhere with agent technologies, it is nearby to think about an integrated, fully agent-based solution, which would be technologically "cleaner", provide a common implementation and communication basis for all parts and possible later extensions, and would open up optimum opportunities for synergy effects between separate functions or OMIS parts.

We did an extensive survey about contemporary agent approaches to OMIS [Elst et al., 2004b; Elst & Abecker, 2004]. The results showed that current systems can be organized along the following dimensions:

- **System development Level:** the question whether agent techniques are used (a) only for organization and requirements analysis; or (b) also for system archtecture design, or (c) really for an implementation based upon multi-agent technology.
- **Macro-level structure of the agent system:** it can be distinguished whether the approach (a) only implements one intelligent agent (typically, for personal assistants); or (b) represents a homogeneous multi-agent system (like many cooperative retrieval systems, all agents are of the same kind); or (c) maintains a heterogeneous agent society containing different types of agents.
- **KM application area:** we can characterize systems according to the question which Mnemonic Function or which KM Processes they support.

In our analysis, we could identify a number of research prototypes and systems which can be considered an agent-based OMIS, or an Agent-Mediated KM system. However, very few systems really aimed at covering large areas of the knowledge lifecycle, were implemented with multi-agent technology, and realized a heterogeneous multi-agent system (this is the configuration which is the most ambitious and promising). To mention the major representatives:

CoMMA. In the CoMMA project [Bergenti et al., 2000] societies of agents are created for personalized information delivery [Gandon & Dieng-Kuntz, 2002]:

- Agents in the *ontology dedicated sub-society* are concerned with the management of the ontological aspects of the information retrieval activity.
- The *annotation dedicated sub-society* is in charge of storing and searching document annotations in a local repository and also of distributed query solving and annotation allocation.
- The *connection dedicated sub-society* provides white page and yellow page services to the agents.
- The *user dedicated sub-society* manages user profiles as well as the interface to the knowledge worker.

The sub-societies in CoMMA can be organized hierarchically or Peer-to-Peer. The position of an agent in a society is defined by its role [Gandon, 2002b]. The system was implemented on top of JADE agent, and special attention was paid to the use of XML and RDF for representing document annotations and queries.

FRODO. The FRODO project which was defined in large parts by the author of this thesis, realizes the OMIS architecture presented here, adopting a multi-agent approach. It is especially dedicated to distributed OMISs. Agents in a FRODO reside on all four layers of the OMIS generic architecture:

- *Workflow-related agents* (task agents, workflow model manager, ...) are on the Application Layer and control the execution of business processes.
- Personal User Agents are also on the Application Layer and provide the interface to the individual knowledge worker.
- On the Knowledge Broker Layer, *Info Agents* and *Context Providers* realize

retrieval and other information processing services to support the task and user agents.

- The knowledge descriptions are handled by *Domain Ontology Agents*.
- Dedicated Distributed *Domain Ontology Agents* serve as bridges between several OMISs.
- *Wrapper Agents* and *Document Analysis and Understanding Agents* enable access to the sources and informal-formal transitions of information, and are thus located in the Knowledge Object Layer or at the intersection between knowledge objects and knowledge descriptions, respectively.

EDAMOK. The Edamok project[127] also aims at enabling autonomous and distributed management of knowledge [Bonifacio et al., 2002a]. Edamok completely abandons centralized approaches, resulting in the *Peer-to-Peer architecture KEx* [Bonifacio et al., 2002b]. Each peer in KEx has the competence to create and organize the knowledge that is local to an individual or a group. Social structures between these peers are established that allow for knowledge exchange between them. In addition to the semantic coordination techniques that are required for this approach, the Edamok project also investigates contextual reasoning, natural language processing techniques and methodological aspects of distributed KM.

It is noticeable that all these three projects came to the conclusion that – for handling the complexity inherent in such distributed KM scenarios – it would make sense to define mechanisms and languages for defining social strcutures between agents. For instance, in FRODO, an agent is not only defined by its Goals, Knowledge, and Competencies (which corresponds roughly to Newell's knowledge level), but also by **Rights** and **Obligations**, that together allow to define **Agent Roles** (cp. [Elst & Abecker, 2002]). Recently, also formal investigations into the theory of such social agent structures have been undertaken [Dignum, 2004]. Altogether, the Agent-Mediated KM topic seems still to provide

[127] http://edamok.itc.it/

thrilling questions, such as:

- *Socio-technical:* How can the teamwork of human knowledge workers and artificial agents (that might act “on behalf of” people) be balanced? Here, questions from human-computer interaction arise, but also questions of trust, responsibility, etc.
- *Agent technology and KM functionality:* What agent models and architectures are needed for what kind of KM application? Should concepts of trust, responsibility, rights, obligations be integrated in the models? How can the flexibility of reactivity and proactivity better be exploited for KM tasks? Which *new* functionalities can agent-based systems offer to KM?
- *Methodological and engineering aspects:* Which functionalities can be provided as a kind of “KM middleware” or as modules for building KM applications? How should agent-orientation of design and implementation be reflected in an “agent-based KM methodology” in order to facilitate transitions between different phases in the development cycle?
- *Evaluation of agent-based KM:* How good does the integration of (not agent-based) legacy systems into agent environments work in real-world applications? How easily can new agent-based components really be integrated into an existing system? What evaluation paradigms can be used to assess agent-based approaches and to make different KM applications more comparable?

5.3 Weakly-Structured Workflow Systems

We will not go in much detail

The topic of flexibility and ad-hoc changes has been discussed in the workflow area for a long time. Further, within the work of the author of this thesis, the topic arose several times, but was never thoroughly elaborated. For these two reasons, we won't go in much detail in this thesis. Nevertheless, we would like to make the point that in spite of the long tradition in flexible workflow, to us it seems still an unsolved problem how knowledge work could be appropriately supported by means of workflow-like tools.

It should be clear from several discussions in this thesis, that real, knowledge-intensive work can hardly be planned in advance to a big extent. Hence, strong-structured process models and workflow approaches seem unsuitable.

Normally, one would suggest to use groupware of CSCW tools which do not expect an explicit process model in advance.

However, seeing a strict separation between these two approaches, seems to be too limited to us:

- On one hand, we would like maximum freedom for changing plans on the fly, for plan refinement during enactment, and for ad-hoc activities.
- On the other hand, one would also like to reuse short sequences of re-occurring activity patterns. Or, embed ad-hoc activities into a strict conventional workflow.

Giving up all functionalities of conventional workflow approaches would mean that our concept of task and process context can hardly survive, that no standardization in any respect, and no experience transfer from prior, similar process instances would be possible.

Hence it would make more sense to design a tool, roughly described as follows:

- A user has to his disposal a library of activity sequences which were earlier useful. This library may be organized along a task ontology which describes the kinds of activities occuring in the given domain of work ([Schwarz, 2003] discusses the idea of task-concept ontologies; the MIT process handbook [Malone et al., 2003] is in some respect similar). Such an activity pattern library should also contain generic information needs as presented in this thesis.

- When being confronted with a new knowledge-intensive task, the user will configure a process model from library patterns, e.g., supported by a retrieval engine which maps characteristics of the actual problem at hand to characteristics of the problems dealt with using the stored process patterns ([Wargitsch, 1997; Wargitsch et al., 1998] presented such an approach using Case-Based Retrieval techniques).
- The user might be supported in constructing and refining his/her process model by support procedures ensuring consistency, quality criteria, etc. ([Rupprecht, 2002] presents a process toolkit that uses current task and environment criteria plus process design rules for helping the user with this process individualisation task).
- Then, during enactment, the user should have the possibility to refine or change on the fly the process model. The system should try to use as much context as possible for knowledge services, regarding both task enactment ("function knowledge") and process improvement ("process knowledge").
- During and after finishing a process instance, the system should try to gather as much feedback as possible in order to improve its knowledge base. To this end, [Wargitsch et al., 1998] used discussion groups and mail contacts between process enacters and process designers, for fostering continuous process improvement. [Holz, 2002] allows to change process model and information needs on the fly and store changed models in the library.

This short description should be enough to get the point. It should also be clear that there are already several really impressing prototypes for different facets of the idea. Nevertheless, there is not yet a fully integrated system, also providing proactive knowledge services. And there is not the slightest evidence that such approaches could become widespread in the near future. Hence, there are obviously some still challenging research questions:

(1) To which extent can the idea of task and process context for

proactive knowledge services be saved, if the process structures become weaker and weaker? Could task ontologies (similar to Web Service registries envisioned in Semantic Web Service scenarios) help to add a new dimension of background knowledge if the proces flow disappears to some extent?

(2) What would be appropriate user interface concepts to make such complex scenarios realistic for "normal" users? In particular when taking into account the high degree of freedom, individuality and creativity that knowledge workers claim.

(3) What are "normal" users for a scenario as we sketched it? Up to now, all approaches going into the sketched direction use engineering application domains (software engineering, mechanical engineering, automotive engieering)?

6 Summary

„I know more than I can tell,
and I can tell more than I can write down."

Dave Snowden (IBM Global Services)

Related work, risks, limitations, and shortcomings, as well as possible future work, have already been discussed extensively in the technical Chapters 3 and 4. Hence we can restrict ourselves here to a short summary of the major contributions of this thesis.

Major Contribution #1: The KnowMore OMIS Reference Architecture

First, we defined a **comprehensive conceptual framework** and a **generic implementation blueprint** for a **Process-oriented Organizational Memory Information System** that realizes **proactive** provision of knowledge services, relying on the notion of **dynamic task context.** Here, especially the utilization of dynamic task context is unique to our approach.

We introduced a **four-layer reference architecture** with an Application Layer, a Knowledge Broker Layer, a Knowledge Description Layer, and a Knowledge Object Layer. We thoroughly discussed possible instantiations of the generic layers and gave plenty of examples for their practical realization. Through the implementation of the KnowMore prototype, we gave a proof-of-concept for the approach. The major general characteristics of our architecture can be summarized as follows:

- Intelligent assistance instead of automated problem-solving
- Extended business process modeling, including context variables
- Expressive, ontology-based Knowledge Item Descriptions, comprising powerful content characterizations, description of knowledge creation and potential usage *context*, as well as virtual knowledge objects.

- Multi-source integration through separate Knowledge Description Layer
- Ontology-based Knowledge Description Layer allows powerful retrieval and processing services
- Open architecture allows manifold later extensions and synergies between functionalities
- Basic approach goes well with widespread standards (in particular, in the workflow area)

Major Contribution #2: The DECOR Total Solution for BPOKM

Second, we designed, implemented, and tested in several case studies, the DECOR **total solution for Business-Process Oriented Knowledge Management** (BPOKM). This solution **comprises** a BPOKM project management approach, a methodological guidance for **process analysis and re-engineering**, a **modelling method and tool**, as well as a process-oriented **knowledge archive** and a **workflow engine** for enactment. We list some remarkable features of our solution:

- Comprehensive method which combines knowledge-oriented task analysis and ontology design
- KM-specific elements can be well integrated with many other contemporary methods for Business Process or Ontology Engineering
- KM-specific task analysis combines elements from best known approaches
- Archive solution based on commercial product; whole approach already close to market
- Pilot applications give evidence for feasibility of combining process and knowledge management and improvement; thorough evaluation is required

Major Contribution #3: Identify Promising Future Research Areas

Third, the work presented in this thesis gave birth to a couple of other interesting research topics besides the main stream of the argumentation followed here. In particular, we discussed:

- **Knowledge Trading on the basis of an extensive Information Ontology.** The topic is in the meanwhile investigated in a running European RTD project. Interesting are (1) the possibility to define a Reference Information Ontology and the question how much effort it is to adapt this for a concrete, new application area; (2) all non-technical aspects, regarding business engineering (pricing, trust, revenue models,

...); (3) evolution aspects for platform content and meta-level knowledge.

- **Agent-Mediated Knowledge Management.** The idea to implement on the basis of multi-agent technologies, support for the whole knowledge lifecycle in a distributed OMIS scenario. The topic was further investigated in the FRODO project and is still gaining increasing interest.

- **Weakly-Structured Workflows**. The idea to implement a flexible tool somewhere in between a (passive, descriptive) project management and a (active, to some extent prescriptive) workflow management approach, leaving the user all freedom for organizing knowledge work on its own, but offering nevertheless contextualized knowledge services.

Further work

There is a whole bunch of further topics which might be interesting future work and were not discussed extensively in this thesis, such as:

More application-oriented challenges:

Long-term evaluations

- Scientifically sound, long-term investigations about the effects of OMIS tools as described in this thesis. To this end, barriers must be overridden (cp. [Sure, 2003]), approapriate evaluation models must be defined interdisciplinarity must be cultivated, and long-term case study partners must be found. One major challenge is the mix of quantitative (such as time spent for information search) and qualitative effects (such as improved or degraded working atmosphere).

E-Government applications

- There is some evidence that e-Government might be a grateful application area (big knowledge differences between involved partners, seldom running processes, need for documentation and "watertight" decisions, relatively formal application domain, reusability of formal ontologies for several purposes, etc.). It would be interesting to find out whether it also provides specific requirements or particular challenges.

Success criteria

- In general, the scenarios where an approach like ours could generate most value-added, are not yet well understood.

More basic-research oriented challenges:

Reference ontologies

- It would make sense to go on with the idea of Reference Information and Domain Ontologies. Up to now, this did not yet really work, although it would have been useful for Expert Systems and nowadays for e-Commerce. Hence there is a need to **better understand promising scenarios and critical success factors**.

Integration with strategy and controlling

- If we emphasize the importance of evaluation (see above), it is a nearby idea to **combine BPOKM evaluation with other methods for Intellectual Capital Measurement** or with metrics for business process performance. From the economic point of view, it would make sense to integrate such BPOKM metrics with strategic planning and controlling (like in the Balanced Scorecard approach), **as well as with standardized service management approaches.**

Ontology-based context similarity

- Finally, it would be a highly interesting problem (from a technical and an application point of view) how to assess **similarity of contexts** in order to determine the context-specific relevance of a Knowledge Object with a description that is such comprehensive as sketched in Section 5.1. (Knowledge Trading example)

7 Appendix: A Run Through the IKA Case

In this appendix we present a demonstration of the DECOR tool as it was used for the IKA case. This might give an idea about the overall look and feel of a business solution as we envision it. The selected business process for the IKA case is the "Granting full old age pension" process already introduced in Section 4.5.

Roles and persons involved.

The process involves three main roles (some others can be neglected for this presentation):

- *Pension secretariat:* responsible for "data entering" tasks and for ensuring the correctness of all the information supplied by the applicant
- *Mover:* responsible for the main tasks of the process, i.e. the examination of all the documents, the decision concerning the applicant's request and the preparation of the decision form
- *Director:* responsible for the final check of the decision and for signing it

User authentication.

When an employee accesses the DECOR tool, he or she is first confronted with the DECOR authentication screen (Figure 97) in order to input user name and password. In our demonstration, the person that first accesses the tool is an employee with administrative privileges in order to create a new process instance.

Figure 97: LoginScreen (left hand) and DECOR Main Window (right hand)

Creation of new process instance.

New process instance

From the main window, the administrator selects the "workflow" tab, where the model of the process is located in order to create the instances (Figure 98).

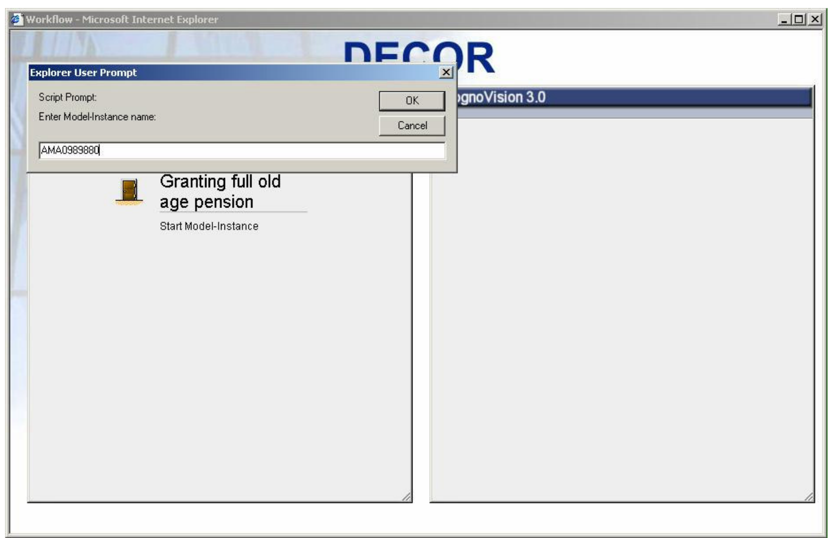


Figure 98: Starting a New Workflow Instance

By pressing the button "Granting full old age pension" (the only process model currently available to this user for starting), the administrator is asked for a name of the instance. In the IKA case, it was useful for archiving reasons to name all the instances of the process after the applicant's social security number which characterises each case.

When the new instance is started, the administrator is informed (see Figure 99), and the first task of the instance is put under the corresponding role's authority.

Figure 99: Instance Successfully Started

Data entering tasks.

Pension Secretariat's tasks

After successful creation of a process instance, IKA employees can use the tool to perform their tasks. The first task belongs to the role "pension secretariat". Such a user (e.g. Ioanna Mitrou) enters the DECOR system via the authentication screen shown above. Then, she is presented with the DECOR main window (Figure 97) and she can access her worklist by pressing the "workflow" button. Figure 100 presents the user's worklist with the first task of the process in it.

Figure 100: Pension Secretariat's Worklist

Fill in Application Form

By pressing the task button, a new window opens which contains the task details and the associated documents (Figure 101).

On the left hand of the task window, the employee has to fill in the values of some variables that are used either for controlling the flow of the process or for searching and retrieving context-specific information in a later process step.

On the right hand of the task window, the user can access the documents that are associated with the specific task. These documents are available either in "read mode" or in "edit mode" depending on the work to be done.

Figure 101: Task "Fill in Application Form "

Application Form

In the current task, the employee has to fill in the application form with the applicant's data as provided by the latter and so, the document is available in "edit mode". By pressing the "Application Form" button, the system calls the respective application (i.e. MS Word) and opens the document for editing. In Figure 102, we see part of the application form the employee should fill in.


ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Ίδρυμα Κοινωνικών Ασφαλίσεων

ΑΙΤΗΣΗ
ΓΙΑ ΑΠΟΝΟΜΗ ΣΥΝΤΑΞΗΣ ΛΟΓΩ ΓΗΡΑΤΟΣ
(ΥΠΟΒΑΛΛΕΤΑΙ ΣΕ ΔΥΟ ΑΝΤΙΓΡΑΦΑ)

Η αίτηση αυτή είναι απλή και συμπληρώνεται εύκολα. Όμως, εάν υπάρχουν δυσκολίες ο ασφαλισμένος μπορεί να συμπληρώσει την αίτηση με τη βοήθεια υπαλλήλου του ΙΚΑ.Η αίτηση υποβάλλεται σε οποιοδήποτε Υποκατάστημα του ΙΚΑ από τον ασφαλισμένο ή από νόμιμο εκπρόσωπο του.

ΠΡΟΣ ΤΟ ΙΚΑ: ΠΕΡΙΦΕΡΕΙΑΚΟ ΑΘΗΝΩΝ	*ΑΡ.ΠΡΩΤΟΚΟΛΛΟΥ/ΗΜΕΡΟΜΗΝΙΑ		*ΚΕΝΤΡΟ ΠΛΗΡΩΜΗΣ
ΠΙΝΑΚΑΣ 1. ΣΤΟΙΧΕΙΑ ΑΣΦΑΛΙΣΜΕΝΟΥ (ΚΕΦΑΛΑΙΑ)			
ΑΡ.ΜΗΤΡΩΟΥ ΙΚΑ 0989880	ΗΜ. ΓΕΝΝΗΣΗΣ	Α.Φ.Μ.	
ΕΠΩΝΥΜΟ ΧΑΡΙΤΟΥ	ΟΝΟΜΑ ΜΑΡΙΑ	ΟΝΟΜΑ ΠΑΤΕΡΑ ΑΝΔΡΕΑΣ	ΟΝΟΜΑ ΜΗΤΕΡΑΣ ΜΑΤΙΝΑ
ΑΡ.ΑΣΤΥΝ.ΤΑΥΤ. Π871075	Δ/ΝΣΗ ΚΑΤΟΙΚΙΑΣ: ΟΔΟΣ-ΑΡΙΘΜΟΣ-Τ.Κ.-ΠΟΛΗ Ή ΧΩΡΙΟ ΧΑΡΙΛΑΟΥ ΤΡΙΚΟΥΠΗ 12, 10682 ΑΘΗΝΑ		ΑΡ.ΤΗΛΕΦΩΝΟΥ
ΧΩΡΑ ΥΠΗΚΟΟΤΗΤΑΣ	ΧΩΡΑ ΓΕΝΝΗΣΗΣ	ΔΗΜ./ΚΟΙΝ.ΓΕΝΝΗΣΗΣ	ΝΟΜΟΣ ΓΕΝΝΗΣΗΣ
ΑΠΑΙΤΟΥΜΕΝΑ ΔΙΚΑΙΟΛΟΓΗΤΙΚΑ: Δ1. Φωτοαντίγραφο αστυνομικής ταυτότητας [ ]			

Figure 102: Application Form

Having entered all the applicant's data in the form, the employee saves it and returns back to the task (Figure 101). The saved form is now located into the knowledge archive under the folder created for the specific instance.

Task completed

When the employee fills in the values of the variables on the left hand of the window, she presses the "finished" button in order the task to be completed. Whenever during the performance of the task, the "cancel" button is pressed, the employee is led back to the worklist but the task is considered to be not completed and thus, remains in the worklist.

Search registry for insured person's data

After the completion of the first task, it is removed from the user's worklist and is replaced by the next task in the process (provided all its preconditions are fulfilled) because it is performed by the same role. The respective task window is depicted in Figure 103.[128] In this task, the employee is asked to search the IKA registry and locate the applicant's data as they exist in IKA files. This search is performed using the applicant's social security number and name which can be seen in the application form by pressing the respective button. No additional data must be entered in the form; for this reason, the document is available only in "show mode". In this mode, the document is converted from its original format to an HTML page and as such it is presented to the user (Figure 104).

[128] For the remainder of the process, in order to focus on the essential things, we will not present the user's worklist anym ore – unless there is something new that needs to be illustrated.

Figure 103: Task "Search registry for insured person's data"

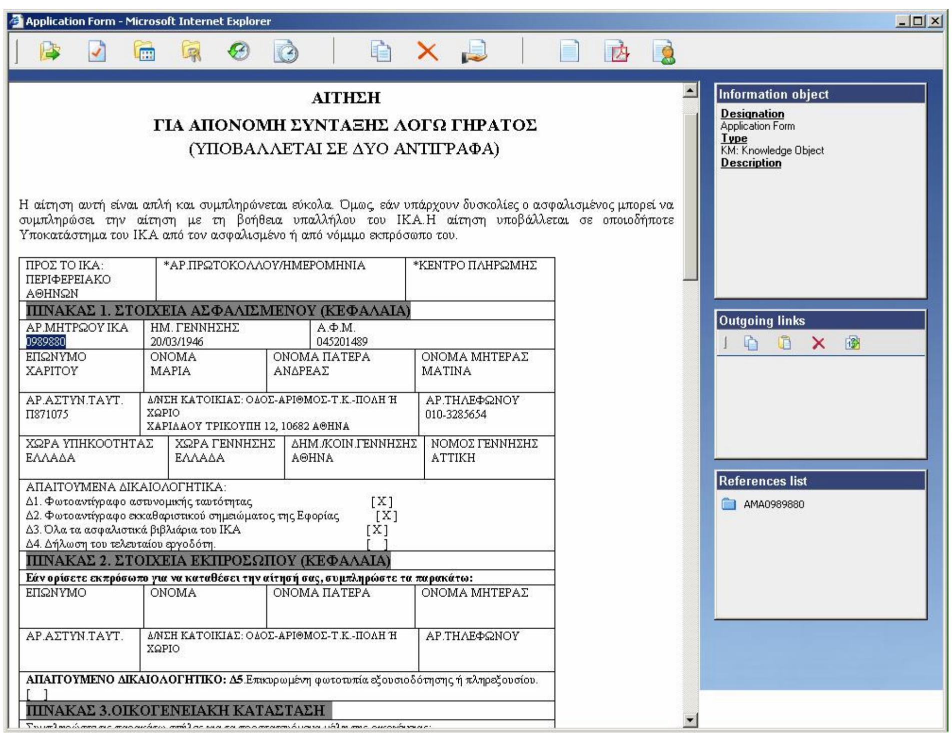


Figure 104: Application Form as HTML Page

Check data When the applicant's data that IKA holds are found, the task can be completed and so, the employee is presented with the next task in her worklist, the details of which are shown in Figure 105.

Figure 105: Task "Check data"

This task involves checking the data entered in the application form (as stated by the applicant) against those that IKA holds for the specific insured member. The output of the task is a decision concerning the correctness of these data.

Assuming that a mismatch exists between the application form and IKA data, the user is presented with an additional task, the task "Correct data", the details of which are shown in Figure 4.12. *Correct data*

Figure 106: Task “Correct data”

The Application Form is opened for editing and the data that do not match IKA data are corrected and the Application Form is saved again. The task is finished by pressing the ”finished” button. Then, the previous task “Check data” is repeated and this loop continues until the user selects “Yes” as the value of the field “Data Correct” (Figure 105). This takes the employee to the last task of the process that is performed by the role “Pension Secretariat”, which is the task “Register Application Form” (Figure 107).

Register Application Form

The meaning of this task is to give the Application Form a unique number for the IKA record. This number is also used by the applicant when contacting IKA to check the status of their application. As in the task “Correct data”, the Application Form is opened for editing and the Record Number is written in the respective field.

Figure 107: Task "Register Application Form"

Pension Secretariat's tasks completed

When the employee completes this task, since there are no other tasks that can be performed by this role, the worklist becomes empty as it is shown in Figure 4.14. However, it is obvious that if other process instances are running, the worklist is not empty but it contains the not completed tasks of the other instances.

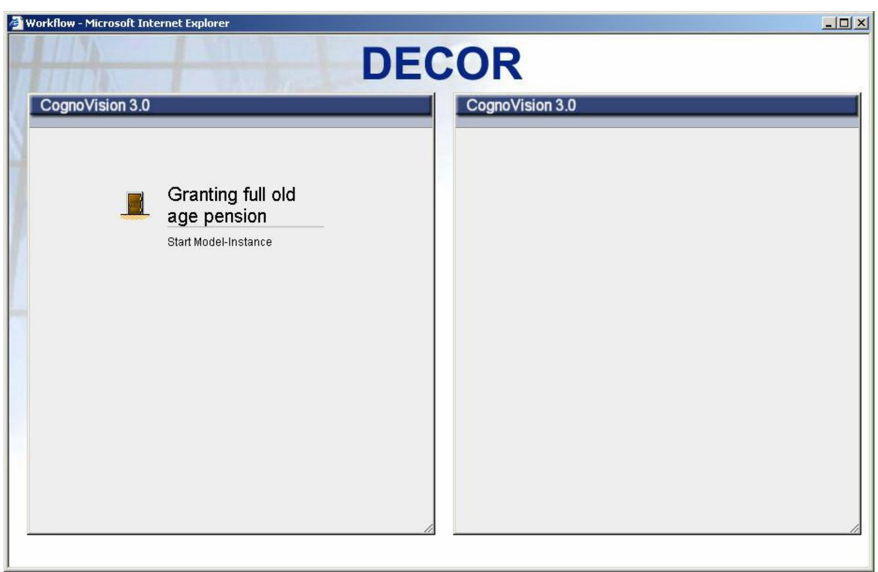


Figure 108: Pension Secretariat's Worklist (empty)

Decision making tasks.

When the tasks performed by the role "Pension Secretariat" are completed, the flow of work passes to an IKA employee that has the right to play the role "Mover". This is the role that accomplishes the main objectives of the process, i.e. the decision about the applicant's request for a pension.

Support for knowledge-intensive tasks

This series of tasks performed by the "Mover" requires a deep knowledge of the relevant legislation; first, for making the decision whether the insured person is entitled to receive a pension; and second, for calculating the amount of pension.

The IKA employee with these knowledge-intensive tasks at hand is supported by the DECOR tool in two ways, as we will see below:

- The system automatically searches, retrieves, and presents the case-specific legal regulations that must be examined in the decision-making tasks.
- Besides the regulations, the user is also presented with the Lessons Learned from similar past cases that other employees have created and stored in the archive.

Coming back to the demonstration of the DECOR tool, let's assume that an employee (e.g. Mrs Nikolopoulou) who is allowed to play the role "Mover" enters the system. The authentication is done in the same way that has already been described.

Mover's worklist

The user is presented with her worklist containing all the tasks that can be performed by her in all running instances. In our case, the task "Examine application and supplementary documentation" is ready to start (Figure 109).

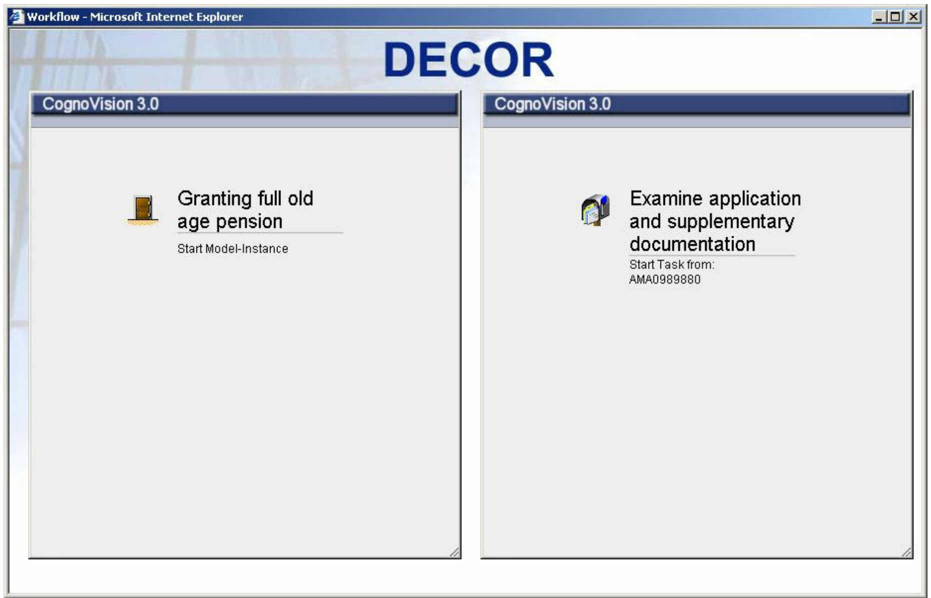


Figure 109: Mover's Worklist

Examine application and supplementary documentation

This task is the first contact of the mover with the insured person's application and logically involves the examination of the application and all the necessary documents that have been supplied by the applicant. As it can be seen in Figure 110, on the left hand of the task window, there are some fields the user must fill in. These are some variables that characterise the case and they will be used later for controlling the flow and for automatically retrieving the case relevant regulations and Lessons Learned. On the right hand of the task window the documents associated with this task are located.

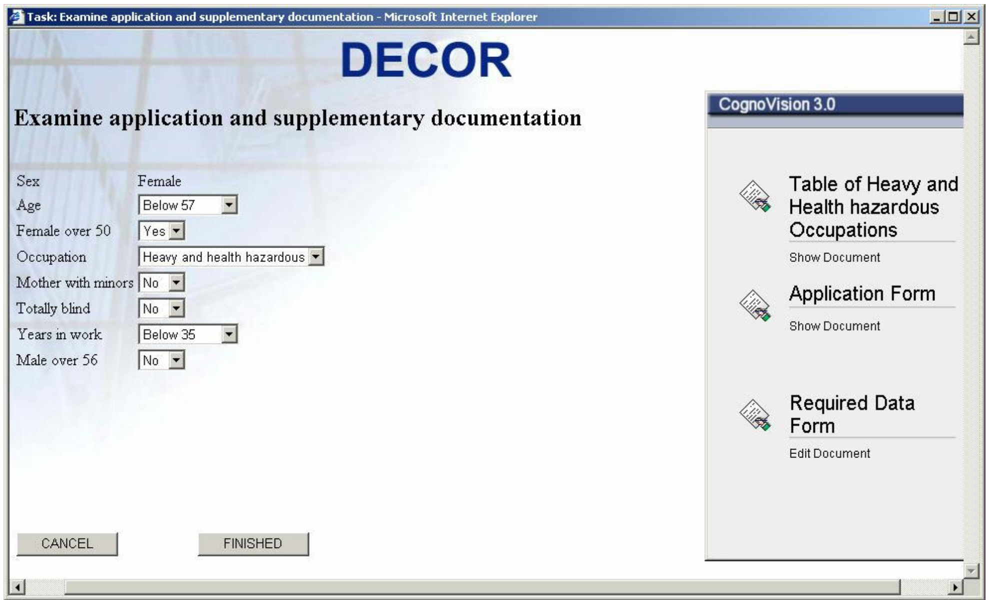
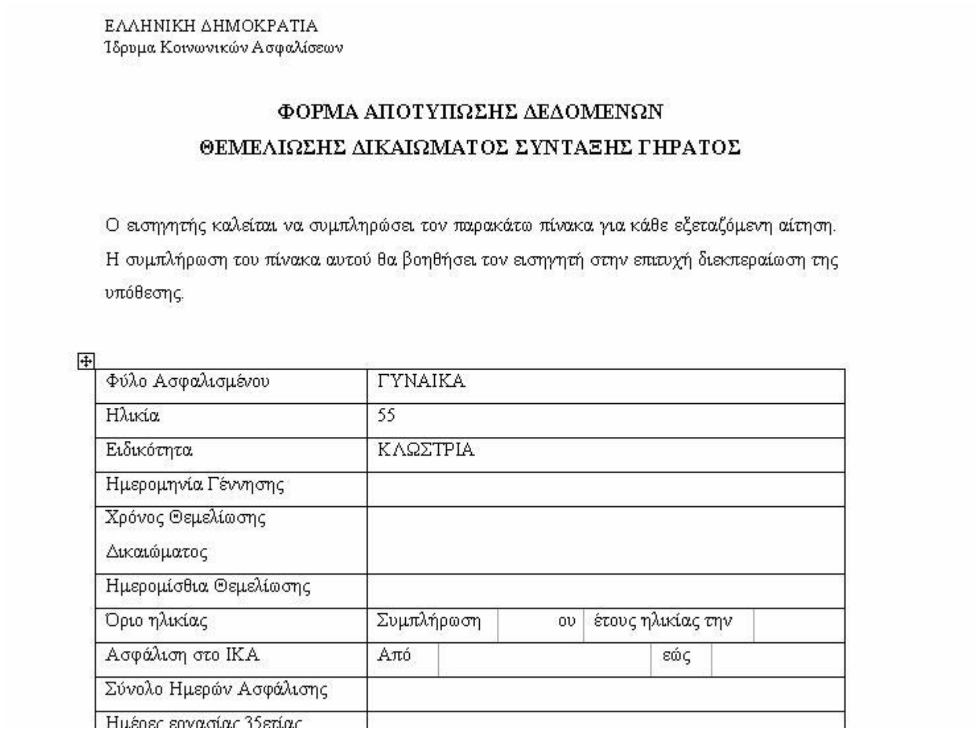


Figure 110: Task "Examine application and supplementary documentation"

Task-relevant documents

These documents include the "Application Form" in show mode, the "Table for heavy and health hazardous occupation" which is a legal document in IKA that contains all the occupations that belong in the category of "heavy and health hazardous". It is used for ensuring that the applicant's occupation belongs to this beneficiary category. Finally, the "Required Data Form" (available in edit mode) is used for summarising all the data concerning the case that are scattered among different documents available in hard copy. It will be used in a later task for reviewing the case and come to a decision. This form is depicted in Figure 111.



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Ίδρυμα Κοινωνικών Ασφαλίσεων

ΦΟΡΜΑ ΑΠΟΤΥΠΩΣΗΣ ΔΕΔΟΜΕΝΩΝ
ΘΕΜΕΛΙΩΣΗΣ ΔΙΚΑΙΩΜΑΤΟΣ ΣΥΝΤΑΞΗΣ ΓΗΡΑΤΟΣ

Ο εισηγητής καλείται να συμπληρώσει τον παρακάτω πίνακα για κάθε εξεταζόμενη αίτηση. Η συμπλήρωση του πίνακα αυτού θα βοηθήσει τον εισηγητή στην επιτυχή διεκπεραίωση της υπόθεσης.

Φύλο Ασφαλισμένου	ΓΥΝΑΙΚΑ				
Ηλικία	55				
Ειδικότητα	ΚΛΩΣΤΡΙΑ				
Ημερομηνία Γέννησης					
Χρόνος Θεμελίωσης Δικαιώματος					
Ημερομίσθια Θεμελίωσης					
Όριο ηλικίας	Συμπλήρωση		ου	έτους ηλικίας την	
Ασφάλιση στο ΙΚΑ	Από		εώς		
Σύνολο Ημερών Ασφάλισης					
Ημέρες εργασίας 35ετίας [illegible]					

Figure 111: Required Data Form

When all the documents are examined or edited and the fields are filled in with the appropriate values stemming from the documentation, the task can be considered completed and the user can press the “Finished” button.

Health Committee-Examine application and issue decision

The next task in the process as it has been modelled using the DECOR modelling tool involves either the examination of the application from a health committee (if a benefit for health condition has been requested and stated in a previous task) or the decision task performed by the mover. In order to have a complete demonstration, let’s assume that such benefit has been requested. So, the application form goes to the “Health Committee” and an employee that can play this role will find into their worklist the task “Examine application and issue decision” the details of which are shown in Figure 112.

The task is finished when the “Health Committee” reaches a decision concerning the applicant’s (or a member of their family) health conditions and issues the respective decision.

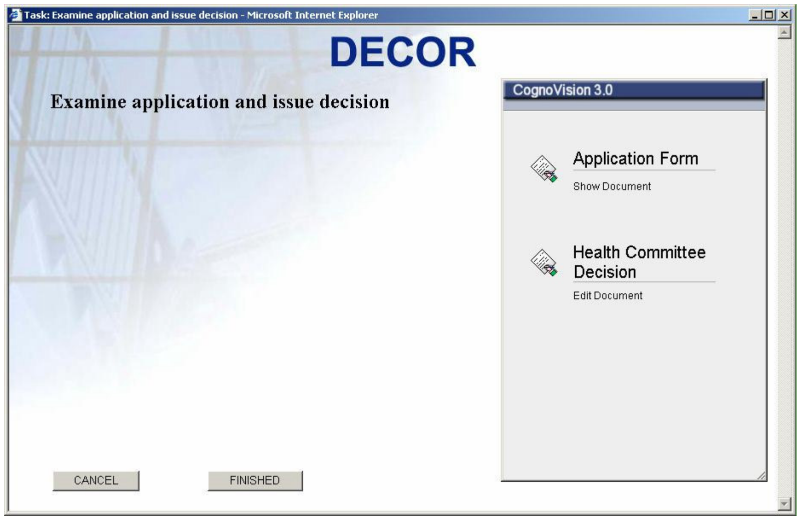


Figure 112: Task "Examine application and issue decision"

Decide about the case

When this task is finished, the flow of work returns back to the mover who has to accomplish the most difficult and knowledge-intensive part of the process, the task "Decide about the case" (see Figure 113).

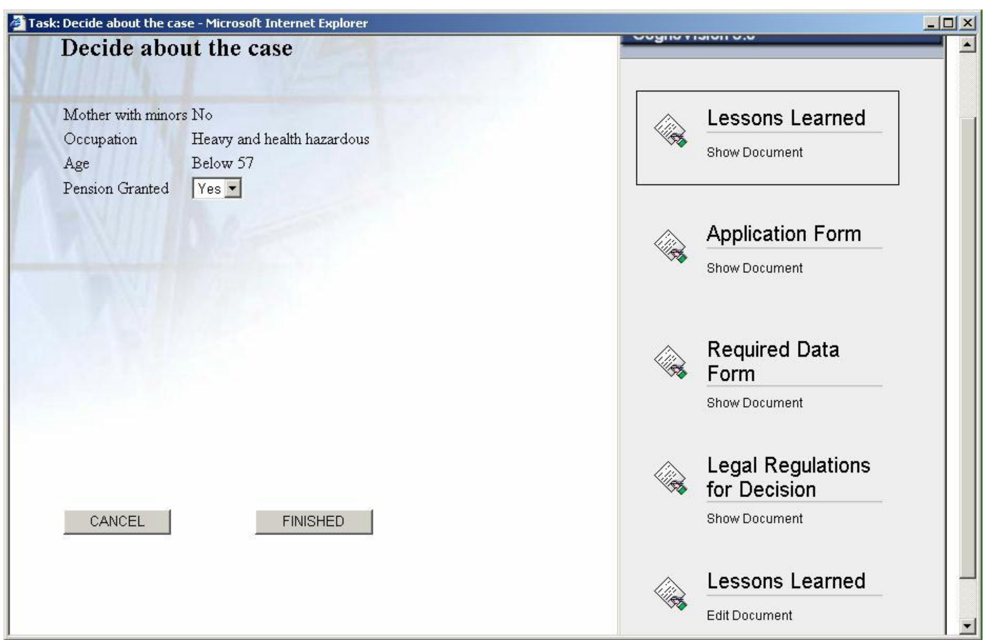


Figure 113: Task "Decide about the case"

As it is shown in Figure 113, the values of some fields that have been filled in a previous task and which characterise the case, are presented on the left side of the task window. If necessary, the employee can find more details in the "Required data form" that has been completed during the mover's previous task and is now available in read mode on the documents' column of the task window.

By pressing the "Legal regulations for Decision" button, the user will have access to all the legal regulations that apply to the case under examination and exist in the DECOR knowledge archive. One of these regulations is displayed for millustration in Figure 114

Retrieval of Legal Regulations

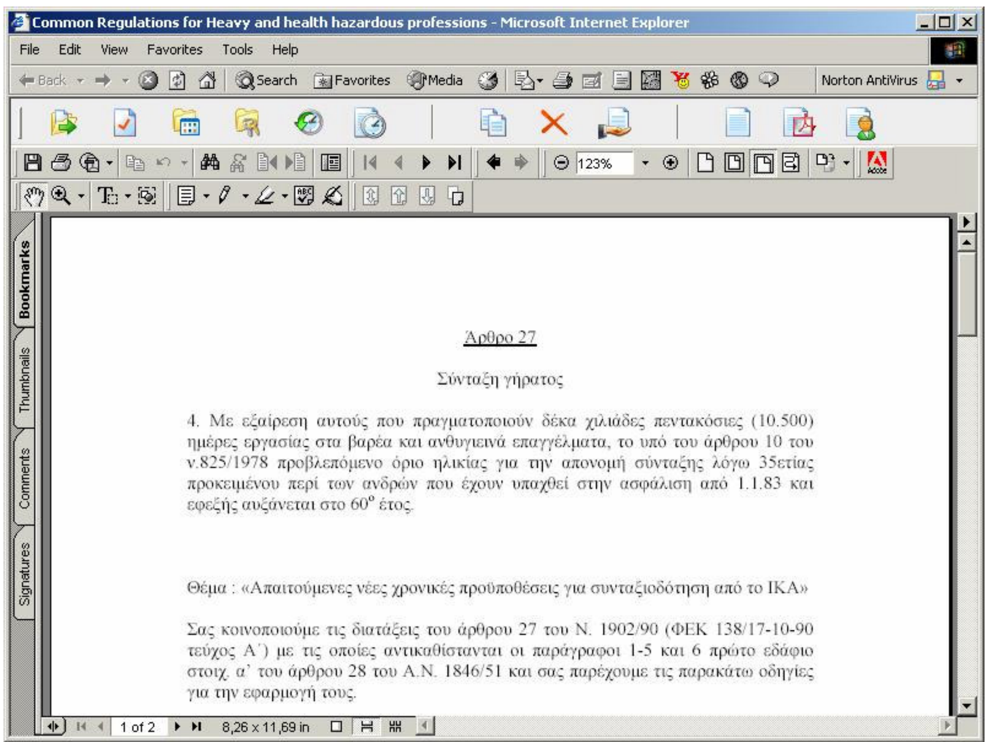
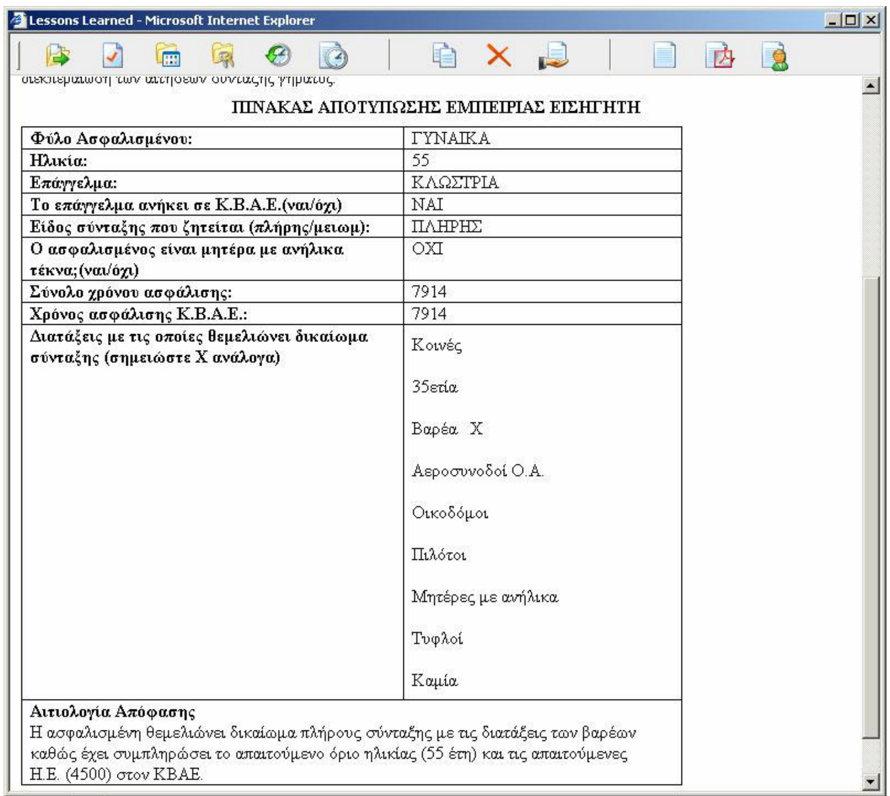


Figure 114: Legal Regulations Relevant for a Decision

Having examined the regulations, the employee may have reached a decision for the applicant's request or may need some additional help. This help is provided from the DECOR tool by pressing the buttons "Lessons Learned". When pressing this button, the system presents the user with all the Lessons Learned documents that other IKA employees have been prepared during the examination of similar cases. The "Lessons Learned" document summarises an examined case containing all the necessary data and also the employee's rational for coming to the decision with references to legal regulations. Such a document is displayed in Figure 115.

Retrieval and Creation of Lessons Learned

Having examined all the regulations that apply to the case at hand and the Lessons Learned from similar past case, the employee should be in a position to come to a decision concerning the insured person's request for a full old age pension. At this point, the employee has the possibility to record their experience for this case by completing the Lessons Learned document which becomes available for editing by pressing the button "Lessons Learned: Edit document". When this button is pressed, the system opens an empty Lessons Learned document, the user fills it in and saves it. When saved, the document is archived by the DECOR system into the knowledge archive and is available for retrieval in future, similar cases.



ΠΙΝΑΚΑΣ ΑΠΟΤΥΠΩΣΗΣ ΕΜΠΕΙΡΙΑΣ ΕΙΣΗΓΗΤΗ

Φύλο Ασφαλισμένου:	ΓΥΝΑΙΚΑ
Ηλικία:	55
Επάγγελμα:	ΚΛΩΣΤΡΙΑ
Το επάγγελμα ανήκει σε Κ.Β.Α.Ε.(ναι/όχι)	ΝΑΙ
Είδος σύνταξης που ζητείται (πλήρης/μειωμ):	ΠΛΗΡΗΣ
Ο ασφαλισμένος είναι μητέρα με ανήλικα τέκνα;(ναι/όχι)	ΟΧΙ
Σύνολο χρόνου ασφάλισης:	7914
Χρόνος ασφάλισης Κ.Β.Α.Ε.:	7914
Διατάξεις με τις οποίες θεμελιώνει δικαίωμα σύνταξης (σημειώστε Χ ανάλογα)	Κοινές 35ετία Βαρέα Χ Αεροσυνοδοί Ο.Α. Οικοδόμοι Πιλότοι Μητέρες με ανήλικα Τυφλοί Καμία

Αιτιολογία Απόφασης
Η ασφαλισμένη θεμελιώνει δικαίωμα πλήρους σύνταξης με τις διατάξεις των βαρέων καθώς έχει συμπληρώσει το απαιτούμενο όριο ηλικίας (55 έτη) και τις απαιτούμενες Η.Ε. (4500) στον ΚΒΑΕ.

Figure 115: Lessons Learned Example

The task "Decide about the case" is completed when the user fills in the field "Pension Granted" with the value that corresponds to their decision, i.e. "yes" or "no".

Calculation of the pension amount

In the case that the decision is negative, the flow of work goes to the task "Issue decision of rejection" whilst if the decision is positive, the flow continues with the task "Calculation of the pension amount". In our example, the decision is positive and therefore, the employee is presented in their worklist with the previously mentioned task, the details of which are shown in Figure 116.

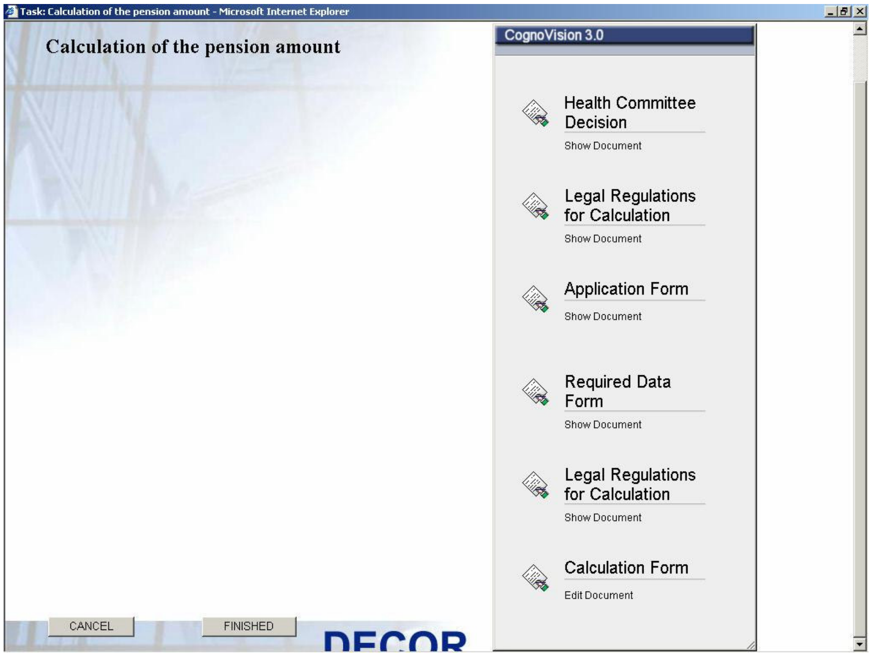


Figure 116: Task "Calculation of the pension amount"

In this task, which is considered to be another knowledge intensive task due to the required knowledge about different regulations, the employee should calculate the exact amount of the pension that the applicant will receive. This is done based on the regulations that apply for each case and the applicant's data concerning days in work, insurable days etc. Moreover, if the applicant has requested a benefit for health reasons, the Health Committee's decision is also taken into account in this task.

Therefore, all the documents are available for displaying in this task, i.e. Application Form, Health Committee's decision (if any) and Required Data form. Moreover, the system using the context variables from previous tasks retrieves and displays the legal regulations for the calculation of the specific pension's amount when the user presses the respective buttons as already described for the task "Decide about the case".

Calculation form

The output of this task is the completion of the "Calculation Form", a document that depicts the different amounts that are taken into account for calculation the final number. This form is opened for edit when the "Calculation Form" button is pressed and it is depicted in Figure 117.

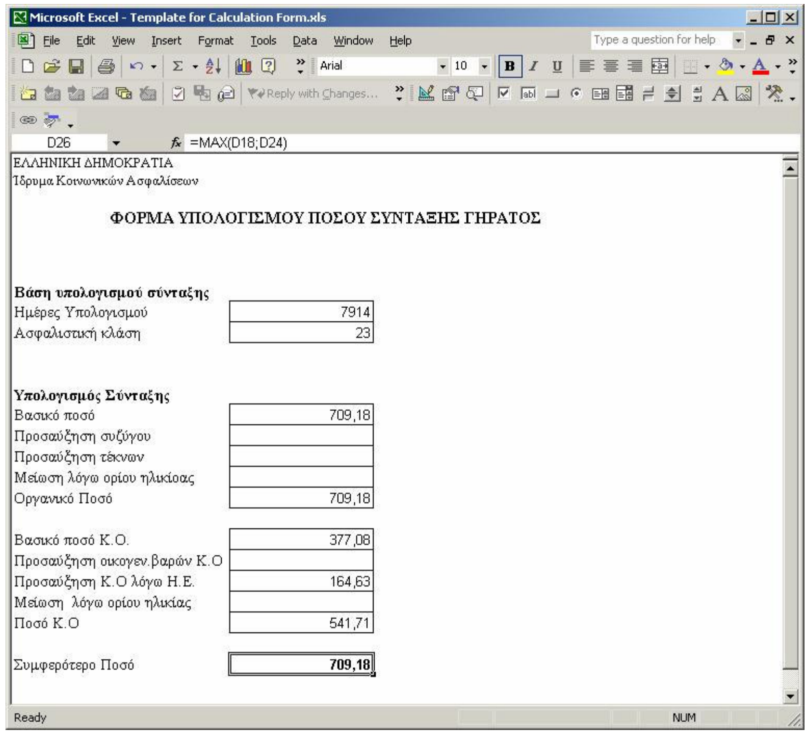


Figure 117: Calculation Form

Issue decision of approval

When the calculation form is filled in and the exact pension amount is calculated, the task is considered to be finished and the mover can start the next task of the process which is the "Issue decision of approval". This task's window is presented in Figure 118.

The output of this task is the preparation of the formal document that describes the decision of approval and which will be later checked and signed by the IKA's branch director.

For the preparation of this decision, the employee has to retrieve data that has already included in the forms "Required data" and "Calculation" in previous tasks. Therefore, these documents are available in this task in show mode.

A template for the "Decision of approval" document is opened for editing when the user presses the respective button and when it is completed and saved the DECOR system archives the document into the knowledge archive.

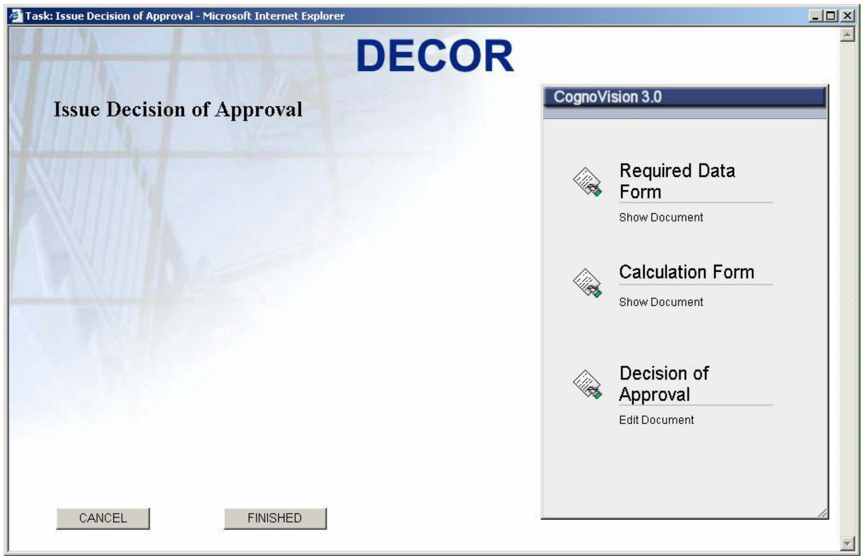


Figure 118: Task "Issue decision of approval"

Decision of Approval

Figure 119 shows part of the Decision of approval for the case presented in this sample run.

Ημερομηνία 20/06/2002

Αριθμός απόφασης 4562

ΑΠΟΦΑΣΗ ΔΙΕΥΘΥΝΤΗ

1. Ο διευθυντής του Υποκαταστήματος αρμόδιος να κρίνει τη με αριθμό πρωτοκόλλου 377-03/01/2002 αίτηση περί Απονομής ΣΥΝΤΑΞΗΣ ΓΗΡΑΤΟΣ ΙΚΑ της ασφαλισμένης με τα κάτωθι στοιχεία:

Α.Μ.Α.: 0989880
Ονοματεπώνυμο: ΧΑΡΙΤΟΥ ΜΑΡΙΑ
Όνομα πατρός: ΑΝΔΡΕΑΣ
Όνομα μητρός: ΜΑΡΙΑ

2. Έχοντας υπόψη τα στοιχεία του φακέλου από τα οποία προκύπτουν τα εξής:

ΑΣΦΑΛΙΣΤΙΚΑ ΣΤΟΙΧΕΙΑ

ΧΡΟΝΟΙ ΑΣΦΑΛΙΣΗΣ ΓΙΑ ΘΕΜΕΛΙΩΣΗ		**ΗΜΕΡΕΣ**
ΚΑΝΟΝΙΚΗ ΑΣΦΑΛΙΣΗ	ΑΠΟ 05/1971 ΕΩΣ 01/2002	7914
ΕΦΕΔΡΙΚΟ ΕΠΙΔΟΜΑ ΕΓΚ.ΙΚΑ34/85	ΑΠΟ ΕΩΣ	
		ΣΥΝΟΛΟ:
ΠΛΑΣΜΑΤΙΚΗ ΑΣΦΑΛΙΣΗ		
ΤΑΚΤΙΚΗ ΑΝΕΡΓΙΑ	ΑΠΟ ΕΩΣ	
		ΣΥΝΟΛΟ:
ΠΡΟΑΙΡΕΤΙΚΗ ΑΣΦΑΛΙΣΗ		
ΠΡΟΑΙΡΕΤΙΚΗ ΣΥΝΕΧΙΣΗ ΑΣΦΑΛΙΣΗΣ	ΑΠΟ ΕΩΣ	

Figure 119: Decision of Approval

In case of a negative decision, the previously described task is named "Issue decision of rejection" and the differences are in the template for the decision document and in the fact that there is no calculation form to be displayed.

Finalisation tasks.

When the Decision document (either positive or negative) has been prepared by the employee that plays the role "Mover", the flow of work passes to the next role involved in the process, the "Director". The Director is responsible for checking the decision and for signing it.

Check and sign decision

Assuming that the IKA's branch director enters the DECOR system using the authentication procedure already described above, they will be presented in their worklist with the task "Check and Sign Decision". In Figure 120, the respective task window is depicted.

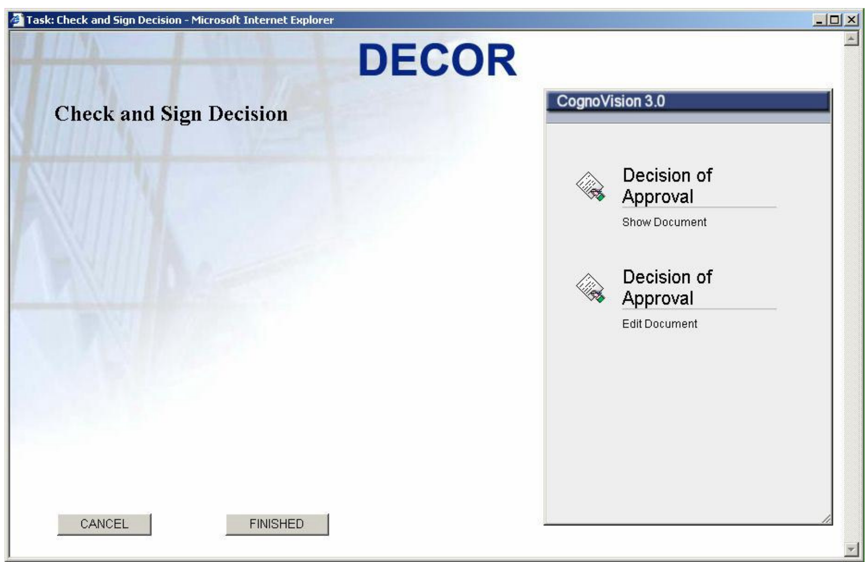


Figure 120: Task "Check and Sign Decision"

The document associated with this task is the "Decision of Approval" (or the "Decision of Rejection" in case of a negative decision) prepared by the Mover in the previous task of the workflow. The document is available in show mode, for reviewing it and in edit mode, in order the director to be able to make changes, print it and sign it.

Notify the applicant

When the Director signs the document, the workflow goes back to the Mover who is informed that the director has signed the Decision. The Mover has to accomplish the final task of the process, i.e. to notify the applicant for the outcome of their request for pension. The details of this task are shown in Figure 121.

The Mover is presented with the final version of the Decision (with possible changes made by the director) and in order the process to be completed, a communication with the applicant must take place.

Figure 121: Task "Notify the applicant"

Process Instances in the DECOR Knowledge Archive.

Process instances in the archive

Every process instance created by a user with administration's privileges corresponds to a new view in the DECOR knowledge archive, created automatically by the DECOR system. Under this view – that carries the name of the process instance – all the documents created during the execution of the instance are stored as Knowledge Objects.

In Figure 122, we see the archive for the completed instances of IKA's process "Granting full old age pension". On the left hand side of the window, the user can browse the folders / views that correspond to the completed process instances. By clicking any of these folders, the user is presented with the documents created during the specific instance on the right window. These documents can be viewed as HTML pages by double clicking on them.

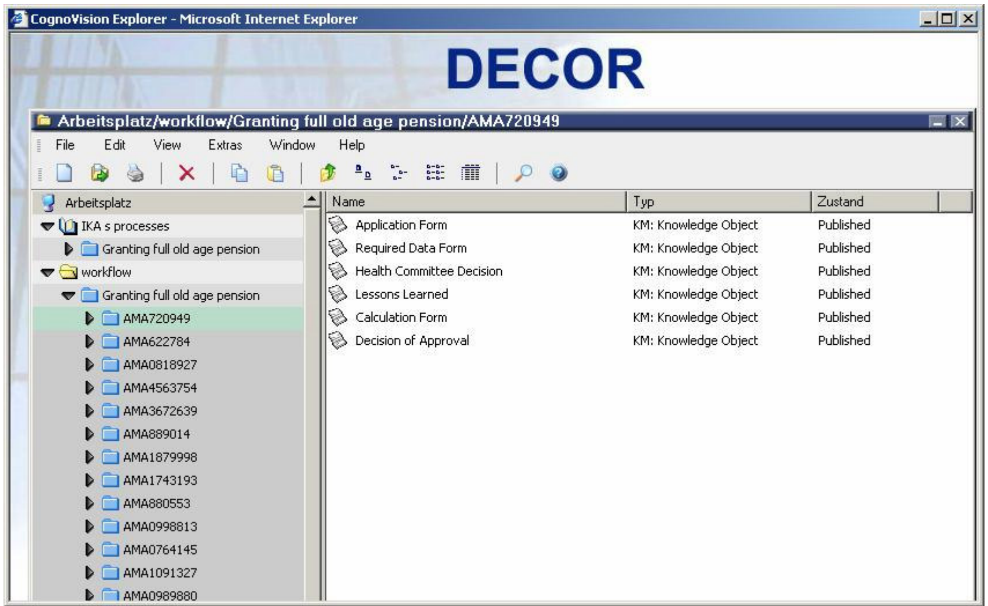


Figure 122: Completed Instances in the DECOR Knowledge Archive

This completed the sample run through an IKA pension granting process, seen from the different perspectives involved.

8 References

[Aalst, 1999]
Aalst, W.M.P. van der (1999): Formalization and Verification of Event-Driven Process Chains. *Information and Software Technology* **41**(10):639-650.

[Aalst, 2002]
Aalst, W.M.P. van der (2002): Making Work Flow: On the Application of Petri Nets to Business Process Management. In: J. Esparza & C. Lakos (eds.): *Applications and Theory of Petri Nets, 23rd Int. Conference ICATPN 2002.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2360, pp. 1-22.

[Aamodt & Plaza, 1994]
Aamodt, A. & Plaza, E. (1994): Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* **7**(1):39-59.

[Aamodt & Nygård, 1995]
Aamodt, A. & Nygård, M. (1995): Different Roles and Mutual Dependencies of Data, Information and Knowledge. *Data & Knowledge Engineering* 16, pp. 191-222, Elsevier.

[Abecker et al., 1997]
Abecker A., Decker, S., Hinkelmann, K. & Reimer, U., eds. (1997): *Workshop „Wissensbasierte Systeme für das Wissensmanagement im Unternehmen“* as part of the 21st German Conference on Artificial Intelligence (KI-97), Freiburg. Document D-97-03. Kaiserslautern: DFKI GmbH.

[Abecker et al., 1998a]
Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O. & Sintek, M. (1998): Towards a Technology for Organizational Memories. *IEEE Intelligent Systems & Their Applications*, **13**(3):40-48. A shortened version of this article was reprinted in: J.W. Cortada (ed.) *Knowledge Management Yearbook, 1999-2000 Edition.* Butterworth Heinemann.

[Abecker et al., 1998b]
Abecker, A., Kühn, O. & Decker, S. (1998): Organizational Memory. *„Das*

aktuelle Schlagwort" im Informatik Spektrum, Springer Verlag. **21**(4):213-214, August 1998. *In German.*

[Abecker et al., 1998c]
Abecker, A., Sintek, M. & Wirtz, H. (1998): From Hypermedia Information Retrieval to Knowledge Management in Enterprises. In: *First International Forum on Multimedia & Image Processing (IFMIP'98).*

[Abecker et al., 1998d]
Abecker, A., Bernardi, A., Hinkelmann, K., Kühn O. & Sintek, M. (1998): *Techniques for Organizational Memory Information Systems*. DFKI Document D-98-02. Kaiserslautern: Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI).

[Abecker et al., 1998e]
Abecker, A., Aitken, S., Schmalhofer, F. & Tschaitschian, B. (1998): KARATEKIT: Tools for the Knowledge-Creating Company. In: *KAW`98, Eleventh Workshop on Knowledge Acquisition, Modeling and Management.*

[Abecker & Decker, 1998]
Abecker, A. & Decker, St. (1998): Topics in Organisational Memory. In: P. Marti & S. Bagnara (eds.): *Designing Collective Memories, 7th Le Travail Humain Workshop, Paris, France.*

[Abecker et al., 1999]
Abecker, A., Bernardi, A. & Sintek, M. (1999): Proactive Knowledge Delivery for Enterprise Knowledge Management. In Ruhe, G. & Bomarius, F. (eds.) *Learning Software Organizations – Methodology and Applications*. Berlin, Heidelberg, New York: Springer Verlag, LNCS 1756.

[Abecker et al., 1999b]
Abecker, A., Bernardi, A. & Sintek, M. (1999b): Enterprise Information Infrastructures for Active, Context-Sensitive Knowledge Delivery. In: *ECIS'99 – The 7th European Conference on Information Systems, Copenhagen, Denmark.* URL:

http://www.dfki.uni-kl.de/~aabecker/Postscript/ECIS99/final-ecis99-word8.htm
Last access: 03/29/2004.

[Abecker & Decker, 1999]
Abecker, A. & Decker, St. (1999): Organizational Memory: Knowledge Acquisition, Integration, and Retrieval Issues. In: F. Puppe (ed.) *XPS-99 / 5. Deutsche Tagung Wissensbasierte Systeme.* Berlin, Heidelberg, New York: Springer-Verlag, LNAI 1570.

[Abecker et al., 2000a]
Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O. & Sintek, M. (2000): Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project. In: [Abecker et al., 2000d], pp. 139-162.

[Abecker et al., 2000b]
Abecker, A., Bernardi, A., Hinkelmann, K., Junker, M., Kühn, O., Sarodnik, T. & Sintek, M. (2000): *KnowMore – Knowledge Managment for Learning Organizations. Final Project Report.* Kaiserslautern: Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI). August 2000.

[Abecker et al., 2000c]
Abecker, A., Bernardi, A., Maus, H., Sintek, M. & Wenzel, C.(2000): Information Supply for Business Processes - Coupling Workflow with Document Analysis and Information Retrieval. *Knowledge-Based Systems* **13**(5):271-284, Special Issue on AI in Knowledge Management, Elsevier.

[Abecker et al., 2000d]
Abecker, A., Decker, S. & Maurer, F., eds. (2000): Knowledge Management and Organizational Memory. *Special Issue of International Journal on Information Systems Frontiers (ISF)* **2**(3/4):139-162, Kluwer.

[Abecker et al., 2000e]
Abecker, A., Bernardi, A., Maus, H. & Wenzel, C. (2000): Information Support for Knowledge-Intensive Business Processes - Combining Workflows With Document Analysis And Information Retrieval. In: St. Staab & D. O'Leary (eds.), *Bringing Knowledge to Business Processes, AAAI Spring Symposium, Stanford, California.* AAAI Technical Report SS-00-03. Menlo Park: AAAI Press.

[Abecker et al., 2001]
Abecker, A., Herterich, R. & Müller, S. (2001): Das DECOR-Projekt – Geschäftsprozeßorientiertes Wissensmanagement mit dem CognoVision-Tool. In: *KnowTech-2001, 3. Kongress über Wissenstechnologien*, Dresden. *In German.*

[Abecker & Mentzas, 2001]
Abecker, A. & Mentzas, G. (2001): Active Knowledge Delivery in Semi-Structured Administrative Processes. In: *Workshop "Electronic Government and Knowledge Management",* Siena / Italy.

[Abecker et al., 2002a]
Abecker, A., Ntioudis, S., Elst, L. van, Houy, C., Legal, M., Mentzas, G., Müller, S. & Papavassiliou, G. (2002): Enabling Workflow-Embedded OM Access With the DECOR Toolkit. In: [Dieng-Kuntz & Matta, 2002], pp..63-74.

[Abecker et al., 2002b]
Abecker, A., Bernardi, A., Ntioudis, S., Herterich, R., Houy, C., Legal, M., Mentzas, G. & Müller, S. (2002): The DECOR Toolbox for Workflow-Embedded Organizational Memory Access. In: J.Filipe, B. Sharp & P. Miranda (eds.), *Enterprise Information Systems III.* Dordrecht: Kluwer Academic Publishers. Pp. 107-116.

[Abecker et al., 2002c]

Abecker, A., Hinkelmann, K., Maus, H. & Müller, H.-J., eds. (2002): *Geschäftsprozessorientiertes Wissensmanagement* Berlin, Heidelberg: Springer-Verlag, Series xpert Press. *In German.*

[Abecker et al., 2003a]
A. Abecker, G. Papavassiliou, S. Ntioudis, G. Mentzas, S. Müller (2003): Methods and Tools for Business-Process Oriented Knowledge Management - Experiences from Three Case Studies. In: F. Weber, K.S. Pawar & K.-D. Thoben (eds.): *ICE 2003 - 9th Int. Conference of Concurrent Enterprising.* Pp. 245-254. Nottingham: Centre for Concurrent Enterprising.

[Abecker et al., 2003b]
A. Abecker, D. Apostolou, W. Maass, G. Mentzas, C. Reuschling, S. Tabor: Towards an Information Ontology for Knowledge Asset Trading. In: F. Weber, K.S. Pawar & K.-D. Thoben (eds.): *ICE 2003 - 9th Int. Conference of Concurrent Enterprising.* Pp. 187-194. Nottingham: Centre for Concurrent Enterprising.

[Abecker et al., 2003c]
Abecker, A., Bernardi, A., Elst, L. van (2003): Agent Technology for Distributed Organizational Memories. In: *ICEIS-2003 - 5th Int. Conference on Enterprise Information Systems.* Pp. 3-10.

[Abecker & Elst, 2003]
Abecker, A. & Elst, L. van (2003): Ontologies for Knowledge Management. In: [Staab & Studer, 2003], pp.435-454 .

[Abecker, 2004]
Abecker, A. (2004): Werkzeuge für das Wissensmanagement – Eine Übersicht. In: F. Mayer, K.-D. Schmitz & J. Zeumer (eds.): *Terminologie und Wissensmanagement - Deutscher Terminologietag 2004.* Pp. 53-67. Köln: SDK Systemdruck GmbH. *In German.*

[Abecker et al., 2004]
Abecker, A., Bernardi, A., Elst, L. van, Klein, B. (2004): Design Principles and Research Directions for Organizational Memory Information Systems in the Global Economy. *Submitted for publication.*

[Acuna & Ferré, 2001]
S.T. Acuña & X. Ferré: Software Process Modelling. In: N. Callaos, I. Nunes da Silva & J. Molero (eds.): *World Multiconference on Systemics, Cybernetics and Informatics, ISAS-SCIs 2001, Vol. I: Information Systems Development.* International Institute of Informatics and Systemics. Pp. 237-242.

[Ackerman, 1994]
Ackerman, M. S. (1994): Augmenting the Organizational Memory: A Field Study of Answer Garden. In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW94).*

[Ackerman & Malone, 1990]
Ackerman, M. S. & Malone, T. W. (1990): Answer Garden: A Tool for Growing Organizational Memory. In: *Proceedings of the ACM Conference on Office Information Systems (COIS'90)*.

[Ackerman & Mandel, 1995]
Ackerman, M. & Mandel, E. (1995): Memory in the Small: An Application to Provide Task-Based Organizational Memory for a Scientific Community. In: *Proc. 28th Annual Hawaii Int. Conference on System Sciences (HICSS-28),* pp. 323-332.

[Ackerman & McDonald, 1996]
Ackerman, M. S. & McDonald, D. W. (1996): Answer Garden 2: Merging Organizational Memory with Collaborative Help. In: *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'96).*

[Ackerman & McDonald, 2000]
Ackerman, M.S. & McDonald, D.W. (2000): Collaborative Support for Informal Information in Collective Memory Systems. In: [Abecker et al., 2000d].

[Agostini et al., 1998]
Agostini, A., De Michelis, G., Jarke, M., Matthes, F., Mylopoulos, J., Pohl, K., Schmidt, J., Woo, C. & Yu, E. (1998): A Three-Faceted View of Information Systems: The Challenge of Change. *Communications of the ACM* **41**(12):64-71.

[Agostini et al., 2003]
Agostini, A., Albolino, A., Boselli, R., Michelis, G. de, Paoli, F. de, Dondi, R. (2003): Stimulating Knowledge Discovery and Sharing. In: *Proc. 2003 Int. ACM SIGGROUP Conference on Supporting Group Work.*

[Akkermans et al., 1999b]
Akkermans, H., Speel, P.-H. & Ratcliffe, A. (1999): Problem, Opportunity, and Feasibility Analysis for Knowledge Management - An Industrial Case Study. *KAW'99, Banff, Canada.*
URL: http://sern.ucalgary.ca/KSI/KAW/KAW99/papers.html
[Last access: 04/18/2004]

[Albrecht, 1993]
Albrecht, F. (1993): *Strategisches Management der Unternehmensressource Wissen.* Frankfurt am Main: Peter Lang Verlag. *In German.*

[Allweyer, 1998]
Allweyer, T. (1998): *Knowledge Process Redesign – Modellierung und Gestaltung der Wissensverarbeitung im Unternehmen.* Whitepaper. Saarbrücken: IDS Prof. Scheer GmbH. *In German.*

[Allweyer, 1998b]
Allweyer, T. (1998): *Gestaltung adaptiver Geschäftsprozesse, Rahmenkonzepte*

und Informationssysteme. Wiesbaden. *In German.*

[Althoff et al., 2001]
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,* Leipzig: Institut für Bildverarbeitung und angewandte Informatik.

[André et al., 1998]
André, E., Rist, T. & Müller, J. (1998): WebPersona: A Life-Like Presentation Agent for the World-Wide Web. *Knowledge-Based Systems* **11**(1):25-36.

[Andreasen et al., 2003]
T. Andreasen, H. Bulskov & R. Knappe (2003): From Ontology over Similarity to Query Evaluation In R. Bernardi & M. Moortgat (eds.): *2nd CoLogNET-ElsNET Symposium - Questions and Answers: Theoretical and Applied Perspectives,* pp. 39-50.

[Andreoli et al., 1995]
Andreoli, J.M., Borghoff, U.M., Pareschi, R. & Schlichter, J.H. (1995): Constraint Agents for the Information Age. *J. Universal Computer Science* **1**(12): 762-789.

[Angus et al., 1998]
Angus, J., Patel J., and Harty, J. (1998): Knowledge Management: Great Concept... But What Is It. *Information Week,* Issue No. 673, March 1998. URL: http://www.informationweek.com. Last access: 29-May-2003.

[Apostolou et al., 2002]
Apostolou D., G. Mentzas , A. Abecker , W-C. Eickhoff , W. Maass , P. Georgolios , K. Kafentzis, S. Kyriakopoulou (2002): Challenges and Directions in Knowledge Asset Trading. In: [Karagiannis & Reimer, 2002], pp 549-564.

[Apostolou et al., 2004]
Apostolou, D., Georgolios, P., Klein, B., Franz, J., Maass, W., Abecker, A., Kafentzis, K., Mentzas, G. (2004): Towards Provision of Knowledge-Intensive Products and Services Over the Web. In: *22nd IASTED International Multi-Conference on Applied Informatics.*

[APQC, 1997]
APQC (1997): *Using Information Technology to Support Knowledge Management, Consortium Benchmarking Study: Final Report.*

[Bach, 1997]
Bach, V. (1997): *Rechnerunterstützung für den Entwurf von Geschäftsprozessen – Anwendungen, Auswahl und Entwicklung von BPR–Tools*. Dissertation der Universität St. Gallen No. 1971. Bamberg: Difo-Druck GmbH. *In German.*

[Bakel et al., 1996]
B. van Bakel, R.T. Boon, N.J. Mars, J. Nijhuis, E. Oltmans & P.E. van der Vet

(1996): *Condorcet Annual Report.* Technical Report UT-KBS-96-12, Knowledge-Based Systems Group, University of Twente.

[Baudin et al., 1992]
C. Baudin, J. Gevins, V. Baya & A. Mabogunje (1992): DEDAL: Using Domain Concepts to Index Engineering Design Information. In: *Proc. Meeting of the Cognitive Science Society, Bloomington, Indiana.*

[Baudin et al., 1995]
Baudin, C., Kedar, S., and Pell, B. (1995): Increasing Levels of Assistance in Refinement of Knowledge-Based Retrieval Systems. In: G. Tecuci & Y. Kodratoff (eds.), *Machine Learning and Knowledge Acquisition – Integrated Approaches.* Academic Press.

[Bauer & Leake, 2001]
Bauer, T. & Leake, D. (2001): Real Time User Context Modeling for Information Retrieval Agents. In: *Proc. 2001 ACM CIKM: Tenth Int. Conf. on Information and Knowledge Management.* ACM.

[Bauer & Leake, 2002]
Bauer, T. & Leake, D. (2002): Calvin: A Multi-Agent Personal Information Retrieval System. In: *Agent Oriented Information Systems 2002: Proc. 4th Int. Bi-Conference Workshop AOIS-2002.*

[Baumann et al., 1997]
Baumann, S., Ben Hadj Ali, M., Dengel, A., Jäger, T., Malburg, M., Weigel, A. & Wenzel, C. (1997): Message Extraction from Printed Documents – A Complete Solution. In: *ICDAR 1997*: 1055-1059.

[Baumann et al., 1997b]
S. Baumann, M. Malburg, H. Meyer auf'm Hofe & Claudia Wenzel (1997): From Paper to a Corporate Memory. In: [Abecker et al., 1997].

[Becerra-Fernandez, 2000]
Becerra-Fernandez, I. (2000): The Role of Artificial Intelligence Technologies in the Implementation of People-Finder Knowledge Management Systems. In: Staab S., Tsui E. & Garner, B. (eds): *Proc. AAAI Spring Symposium Bringing Knowledge to Business Processes.*

[Becerra-Fernandez, 2001]
Becerra-Fernandez, I. (2001): Locating Expertise at NASA. *Knowledge Management Review* **4**(4):33-37.

[Becker et al., 1999]
Becker, J., Uthmann, C., Muehlen, M. zur, Rosemann, M. (1999): Identifying the Workflow Potential of Business Processes. In: *Proc. 32nd Hawaii International Conference on System Sciences.* IEEE.

[Benjamins et al., 2002]
Benjamins V.R., Cobo J.M.L., Contreras J., Casillas J., Blasco J., Otto B. de, Garcia J., Blázquez M. & Dodero J.M. (2002) : Skills Management in Knowledge

Intensive Organizations. In: Gomez-Perez, A. & Benjamins V.R. (eds.): *Knowledge Engineering and Knowledge Management. 13th Int. Conf. (EKAW 2002).* Berlin, Heidelberg, New York: Springer-Verlag.

[Berendt et al., 2004]
Berendt, B., Stumme, G., & Hotho, A. (in press). Usage Mining for and on the Semantic Web. In : H. Kargupta, A. Joshi, K. Sivakumar, & Y. Yesha (eds.), *Data Mining: Next Generation Challenges and Future Directions.* Menlo Park, CA: AAAI/MIT Press.

[Bergmann & Schaaf, 2003]
Bergmann, R. & Schaaf, M. (2003): Structural Case-Based Reasoning and Ontology-Based Knowledge Management: A Perfect Match*? Journal of Universal Computer Science* **9**(7):608-626.

[Bergenti et al., 2000]
Bergenti, F., Poggi, A. & Rimassa, G. (2000): Agent Architectures and Interaction Protocols for Corporate Management Systems. In: *Proc. ECAI-2000 Workshop on Knowledge Management and Organisational Memories,* pp. 39-47.

[Bernaras et al., 1996]
Bernaras, A., Laresgoiti, I. & Corera, J. (1996) : Building and Reusing Ontologies for Electrical Network Applications. In: *Proc. European Conference on Artificial Intelligence (ECAI-96).*

[Bernardi, 1997]
Bernardi, A. (1997): Electronic Fault Recording: A Corporate Memory for Maintenance Support of Complex Machines. In: *ISMICK'97 International Symposium on the Management of Industrial and Corporate Knowledge, Compiègne.*

[Bernardi et al., 1998]
Bernardi, A., Sintek, M. & Abecker, A. (1998): Combining Artificial Intelligence, Database Technology, and Hypermedia for Intelligent Fault Recording. In: *Sixth International Symposium on Manufacturing with Applications (ISOMA'98), Anchorage, Alaska, USA.* May 1998. This paper received the ISOMA-98 Best Paper Award.

[Bernardi et al., 1998a]
Bernardi, A., Hinkelmann, K. & Sintek, M. (1998): Information Systems in Knowledge Management: An Application Example. In: *Practical Applications in Knowledge Management, PAKeM'98*, Blackpool / UK. The Practical Application Company.

[Bernardi et al., 1998b]
Bernardi, A., Hinkelmann, K. & Sintek, M. (1998): Model-Based Information Systems for Knowledge Management. In: *IT&Knows, Conference on Information Technology and Knowledge Systems, a part of the 15th IFIP World Computer Congress*, Vienna & Budapest, August 1998.

[Berners-Lee et al., 2001]

Berners-Lee, T., Hendler, J. & Lassila, O. (2001): The Semantic Web. *Scientific American,* May 2001.

[Bertalozzi et al., 2001]
Bertalozzi, B., Krusich, C., Missikoff, M. (2001): An Approach to the Definition of a Core Enterprise Ontology: CEO. In: *OES-SEO2001 – Int. Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations.* URL: http://cersi.luiss.it/oesseo2001/ . [Last access: 03/30/2004].

[Bläsius et al., 1997]
Bläsius, K.-H., Grawemeyer, B., John, I. & Kuhn, N. (1997): Knowledge-Based Document Analysis. In: *4th Int. Conference Document Analysis and Recognition (ICDAR '97).*

[Blessing et al., 2001]
Blessing, D., Görk, M. & Bach, V. (2001): Management of Customer and Project Knowledge: Solutions and Experience at SAP. *J. Knowledge and Process Management.* **8**(2):75-90. John Wiley & Sons, Inc.

[Böhm & Schulze, 1995]
Böhm, M. & Schulze, W. (1995): Grundlagen von Workflow-Management-systemen. *Wissenschaftliche Beiträge zur Informatik* **8**(2):50-65. Dresden: TU Dresden. *In German.*

[Bolchini & Mylopoulos, 2003]
Bolchini, D. & Mylopoulos, J. (2003): From Task-Oriented to Goal-Oriented Web Requirements Analysis. In: *4th Int. Conference on Web Information Systems Engineering (WISE 2003).* Los Alamitos: IEEE Computer Society. Pp. 166-175.

[Bonifacio et al., 2002a]
Bonifacio, M., Bouquet, P., Mameli, G. & Nori, M. (2002): Kex: A Peer-to-Peer Solution for Distributed Knowledge Management. In: [Karagiannis & Reimer, 2002], pp. 490-500.

[Bonifacio et al., 2002b]
Bonifacio, M., Bouquet, P. & Traverso, P. (2002): Enabling Distributed Knowledge Management. Managerial and Technological Implications. *Novatica Informatik/Informatique* **3**(1):23-29.

[Borghoff & Pareschi, 1998]
Borghoff, U.M. & Pareschi, R., eds. (1998): *Information Technology for Knowledge Management*. Berlin, Heidelberg, New York: Springer-Verlag.

[Borghoff & Schlichter, 2000]
Borghoff, U.M. & Schlichter, J. (2000): *Computer-Supported Cooperative Work. Introduction to Distributed Applications.* Berlin, Heidelberg, New York: Springer-Verlag.

[Borowsky et al., 1998]
Borowsky, R., Hofer-Alfeis, J., Klabunde, S., Schneider, A., Schmidt R. & Schoen, S. (1998): *Lösungsbausteine für Wissensmanagement-Prozesse.*

[Brand et al., 2003]
Brand, A., Diefenbruch, M., Schaal, H.-G. & Hoffman, M. (2003): Intelligente Wissensportale - Neue Lösung für prozessorientiertes Wissensmanagement. Online Article from ExperPraxis 2003/2004, 10.03.2003. URL http://www.experteam.de/startd/publikationen/Artikel/ [Last access: 04/17/2004]

[Brézillon et al., 2000a]
Brézillon, P., Naveiro, R., Cavalcanti, M. & Pomerol, J.-Ch. (2000): SART: An Intelligent Assistant System for Subway Control. *Pesquisa Operacional* **20**(2):247-268.

[Brézillon et al., 2000b]
Brézillon, P., Pasquier, L. & Pomerol, J.-Ch. (2000): SART: Reasoning with Contextual Graphs. *European Journal of Operational Research* **136**(2):290-298. Elsevier.

[Brown, 1996]
Brown, L.A. (1996): *Designing and Developing Electronic Performance Support Systems.* Digital Press.

[Buckingham Shum, 1997]
Buckingham Shum, S. (1997): Negotiating the Construction and Reconstruction of Organisational Memories. *Journal of Universal Computer Science* **3**(8):899-928. Also appeared in [Borghoff & Pareschi, 1998].

[Buckingham Shum, 1997b]
Buckingham Shum, S. (1997b): Balancing Formality with Informality: User-Centred Requirements for Knowledge Management Technologies. AAAI Spring Symposium on AI in Knowledge Management, Stanford, CA. Menlo Park: AAAI Press.
URL: http://kmi.open.ac.uk/~simonb/org-knowledge/aikm97/sbs-paper1.html
[Last access: 04/25/2004].

[Budzik & Hammond, 1999]
Budzik, J. & Hammond, K.J. (1999): Watson: Anticipating and Contextualizing Information Needs. In: *Proceedings of the Sixty-second Annual Meeting of the American Society for Information Science*.

[Budzik & Hammond, 2000]
Budzik, J. & Hammond, K.J. (2000): User Interactions with Everyday Applications as Context for Just-in-time Information Access. In: *Proceedings of Intelligent User Interfaces 2000.* ACM Press.

[Budzik et al., 2001]
Budzik, J., Hammond K. & Birnbaum, L. (2001): Information Access in Context. *Knowledge-Based Systems* **14**(1-2):37-53. Elsevier Science.

[Budzik et al., 2002]
Budzik, J., Bradshaw, S., Fu, X. & Hammond, K. (2002): Supporting Online Resource Discovery in the Context of Ongoing Tasks with Proactive Software Assistants. *International Journal of Human-Computer Studies* **56**(1):47-74.

[Bullinger et al., 1997]
Bullinger, H.-J., Wörner, K. & Prieto, J. (1997): *Wissensmanagement heute – Daten, Fakten, Trends.* Stuttgart: Fraunhofer Institut für Arbeitswirtschaft und Organisation IAO. *In German.*

[Bullinger et al., 1998]
Bullinger, H.J., Warschat, J., Prieto, J. & Wörner, K. (1998): Wissensmanagement – Anspruch und Wirklichkeit: Ergebnisse einer Unternehmensstudie in Deutschland. *Information Management* 1/98, pp. 7-23. *In German.*

[Burstein et al., 1998]
Burstein, F., Linger, H., Stein, E. & Jennex, M. (1998): Researching Organisational Memory Systems. *Presentation at 31st Annual Hawaii Int. Conference on System Sciences (HICSS-31).*

[Carbonell, 1996]
Carbonell, J. (1996): Digital Librarians: Beyond the Digital Book Stack. *IEEE Intelligent Systems* **11**(3).

[Cattell & Barry, 2002]
Cattell, R.G.G. & Barry, D.K., eds. (1997): *The Object Database Standard: ODMG 2.0.* Morgan Kaufmann Publishers.

[Celentano et al., 1995]
Celentano, A., Fugini, M.G. & Pozzi, S. (1995): Knowledge-based Document Retrieval in Office Environments: The Kabiria System. *ACM Transactions on Information Systems (TOIS)* **13**(3):237-268. New York: ACM Press.

[Chu et al., 1996]
Chu, W.W., Yang, H., Chiang, K., Minock, M., Chow, G. & Larson, C. (1996): CoBase: A Scalable and Extensible Cooperative Information System. *Journal of Intelligent Information Systems* **6**(2-3):223-259. Boston: Kluwer Academic Publishers.

[Cimiano et al., 2004a]
Cimiano, P., Handschuh, S. & Staab, S. (2004): Towards the Self-annotating Web. In: *13th Int. World Wide Web Conference, WWW 2004. To appear.*

[Cimiano et al., 2004b]
Cimiano, P., Ehrig, M., Gabel, T., Haase, P., Hefke, M. & Stojanovic, N. (2004): Similarity for Ontologies – A Comprehensive Framework. *Submitted for Publication.*

[Cohen & Prusak, 2001]
Cohen, D. & Prusak, L. (2001): *In Good Company: How Social Capital Makes Organizations Work.* Harvard Business School Press.

[Cole et al., 1997]
Cole, K., Fischer, O. & Saltzman, P. (1997): Just-in-time knowledge delivery. *Communications of the ACM* **40**(7):49-53. New York: ACM Press. Online available under URL: http://doi.acm.org/10.1145/256175.256184 .
[Last access: 02/15/2004].

[Conklin & Weil, 1997]
Conklin, E. J. & Weil, W. (1997): *Wicked Problems: Naming the Pain in Organizations.* A White Paper by Group Decision Support Systems, Washington DC. URL: http://www.leanconstruction.org/pdf/wicked.pdf
[Last access: 03/29/2004].

[Dämmig et al., 2002]
Dämmig, I., Hess, U. & Borgmann, C. (2002): Kommunikationsdiagnose (KODA) – Einstiegsmethode und –werkzeug in das praktische Wissensmanagement. In: [Abecker et al., 2002c], pp. 123-158. *In German.*

[Davenport et al., 1996]
Davenport, Th.H., Jarvenpaa, S.L. & Beers, M.C. (1996): Improving Knowledge Work Processes. *Sloan Management Review*, Reprint Series **37**(4), Summer, 1996.

[Davenport & Prusak, 1998]
Davenport, Th. H. & Prusak, L. (1998): *Working knowledge – how organizations manage what they know.* Boston: Harvard Business School.

[Davenport, 2002]
Davenport, T.H. & Glaser, J. (2002): Just-in-Time Delivery Comes to Knowledge Management. *Harvard Business Review* **80**(7):107-111.

[Decker et al., 1995]
Decker, K., Lesser, V., Nagendra Prasad, N.V. & Wagner, T. (1995): MACRON: An Architecture for Multi-agent Cooperative Information Gathering. In: *Proc. CIKM '95 Workshop on Intelligent Information Agents.*

[Decker et al., 1999]
Decker, S., Erdmann, M., Fensel, D. & Studer, R. (1999): Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In R. Meersman et al. (eds.): *Semantic Issues in Multimedia Systems*, pp. 351-369. Boston: Kluwer Academic Publishers.

[Decor, 2002]
Decor Project Team (2002): *DECOR Deliverable D28 – Final Public Report.* Kaiserslautern: DFKI GmbH.

[Dengel et al., 2002]
Dengel, A., Abecker, A., Bernardi, A., Elst, L. van, Maus, H., Schwarz, S. & Sintek, M. (2002): Konzepte zur Gestaltung von Unternehmensgedächtnissen. *KI – Künstliche Intelligenz* , 1/02, pp. 5-11. Bremen: arendtap Verlag. *In German.*

[Denning, 2000]
Denning, St. (2000): *The Springboard: How Storytelling Ignites Action in Knowledge-Era Organizations.* Butterworth-Heinemann.

[Diefenbruch et al., 2002]
Diefenbruch, M., Goesmann, T., Herrmann, T. & Hoffmann, M. (2002): Kontext-Navigator und ExperKnowledge – Zwei Wege zur Unterstützung des Prozesswissens in Unternehmen. In: [Abecker et al., 2002], pp. 275-291. *In German.*

[Dieng-Kuntz & Matta, 2002]
Dieng-Kuntz, R. & Matta, N., eds. (2002): *Knowledge Management and Organizational Memories.* Dordrecht, Norwell, New York, London: Kluwer Academic Publishers.

[Dieng & Vanwelkenhuysen, 1996]
Dieng, R. & Vanwelkenhuysen, J., eds. (1996): *Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW`96), Special Track on Corporate Memory and Enterprise Modeling.*

[Dignum, 2004]
Dignum, V. (2004): *A Model for Organizational Interaction: Based on Agents, Founded in Logic.* Ph.D. Thesis, Utrecht University.

[DIN 66201]
Deutsches Institut für Normung e.V., ed. (1981): *DIN 66201: Prozeßrechensysteme.* Berlin: Beuth Verlag GmbH.

[Drucker, 1989]
Drucker, P. F. (1989): *The New Realities.* New York: HarperCollins Publishers.

[Drucker, 1999]
Drucker, P.F. (1999): Knowledge-Worker Productivity: The Biggest Challenge. *California Management Review* **41**(2):79-94.

[Eliassi-Rad, 2001]
Eliassi-Rad, T. (2001): *Building Intelligent Agents that Learn to Retrieve and Extract Information.* PhD thesis, Department of Computer Sciences, University of Wisconsin-Madison.

[Elst & Abecker, 2001]
Elst, L. van & Abecker, A. (2001): Integrating Task, Role, and User Modeling in Organizational Memories. In: I. Russell, & J. Kolen (eds.): *Fourteenth Int. FLAIRS Conference.* Pp. 295-99, Menlo Park: AAAI Press.

[Elst & Abecker, 2002a]
Elst, L. van & Abecker, A. (2002): Domain Ontology Agents for Distributed Orga-

nizational Memories. In: [Dieng-Kuntz & Matta, 2002], pp. 147-158.

[Elst & Abecker, 2002b]
Elst, L. van & Abecker, A. (2002): Ontologies for Information Management: Balancing Formality, Stability, and Sharing Scope. *Expert Systems with Applications* **23**(4):357-366. Elsevier.

[Elst & Abecker, 2004]
Elst, L. van & Abecker, A. (2004): Agent-based Knowledge Management. *KI – Künstliche Intelligenz, Special Issue "Anwendungen von Software Agenten". To appear Spring 2004.* Bremen: arendtap Verlag.

[Elst et al., 2001]
Elst, L. van, Abecker, A. & Maus, H. (2001): Exploiting User and Process Context for Knowledge Management Systems. In: *Workshop on User Modelling for Context-Aware Applications* at the User Modeling Conference 2001 (UM-2001).

[Elst et al., 2003]
Elst, L. van, Aschoff, F.-R., Bernardi, A., Maus, H. & Schwarz, S. (2003): Weakly-structured Workflows for Knowledge-intensive Tasks: An Experimental Evaluation. In: *Proc. Twelfth IEEE Int. Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2003),* pp. 340-345. IEEE Press.

[Elst et al., 2004a]
Elst, L. van, Abecker, A., Bernardi, A., Lauer, A., Maus, H. & Schwarz, S. (2004): An Agent-Based Framework for Distributed Organizational Memories. In: M. Bichler, C. Holtmann, S. Kirn, J. P. Müller, C. Weinhardt (eds.): *Coordination and Agent Technology in Value Networks, Multikonferenz Wirtschaftsinformatik (MKWI-2004),* pp. 181-196. Berlin: GITO-Verlag.

[Elst et al., 2004b]
Elst, L. van, Dignum, V., Abecker, A., eds. (2004): *Agent Mediated Knowledge Management.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2926.

[Elst & Schmalhofer, 1999]
Elst, L. van & Schmalhofer, F. (1999): A Cooperative Comprehension Assistant for Intranet-Based Information Environments. In: M. Klusch, O. M. Shehory & G. Weiss (eds.), *Cooperative Information Agents III, Proc. Third Int. Workshop, CIA'99,* pp. 390-401. Berlin, Heidelberg, New York: Springer-Verlag.

[Eppler et al., 1999]
Eppler, M., Seifried, P. & Röpnack, A. (1999): Improving Knowledge Intensive Processes through an Enterprise Knowledge Medium. In: Prasad, J. (ed.): *Proc. ACM SIGCPR Conf. on Managing Organizational Knowledge for Strategic Advantage.*

[Eseryel et al., 2002]
Eseryel, D., Ganesan, R. & Edmonds, G.S. (2002): Review of Computer-Supported Collaborative Work Systems. *Educational Technology & Society* **5**(2): 130-136. April 2002. IEEE Learning Technology Task Force. Online journal

available under URL: http://ifets.ieee.org/periodical/ . [Last access: 02/15/2004].

[Fensel, 1997]
Fensel, D. (1997): An Ontology-Based Broker: Making Problem-Solving Methods Reuse Work. In: *Proc. IJCAI-97 Workshop on Problem-Solving Methods for Knowledge-Based Systems.*

[Fensel, 1998]
Fensel, D. (1998): *Understanding, Developing and Reusing Problem-Solving Methods.* Habilitation in Angewandter Informatik, Fakultät für Wirtschaftswissenschaften, Universität Karlsruhe.,

[Ferstl & Sinz, 1993]
Ferstl, O. & Sinz, E. (1993): *Der Modellansatz des Semantischen Objektmodells SOM.* Bamberger Beiträge zur Wirtschaftsinformatik, Nr.18, Universität Bamberg. *In German.*

[Finkelstein et al., 1994]
Finkelstein, A., Kramer, J. & Nuseibeh, B. (1994): *Software Process Modelling and Technology*. Research Studies Press (distributed by Wiley).

[Fillies et al., 2001
Fillies, C., Weichhardt F. & Koch-Süwer, G. (2001): Prozessmodellierungswerkzeuge und das Semantic Web. In: [Müller et al., 2001]. *In German.*

[Fischer et al., 1991]
Fischer, G., Lemke, A.C., Mastaglio, T. & Morch, A.I. (1991): The Role of Critiquing in Cooperative Problem Solving. *ACM Transactions on Information Systems (TOIS)* **9**(2):123-151. New York: ACM Press.

[Förtsch, 1996]
Förtsch, Ch. (1996): Information Filtering and Information Retrieval in Engineering – The PRISE Project. In: Wolf, M. & Reimer, U. (eds.): *Proc. PAKM 1996, First Int. Conference on Practical Applications of Knowledge Management.*

[Förtsch, 1998]
Förtsch, Ch. (1998): *Kontextmodellierung als Schnittstelle von Hypertext zu Information Retrieval.* Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg. *In German.*

[Fox et al., 1995]
Fox, M.S., Barbuceanu, M. & Gruninger, M. (1995) : An Organisation Ontology for Enterprise Modelling – Preliminary Concepts for Linking Structure and Behaviour.

[Fox & Gruninger, 1998]

Fox, M.S. & Gruninger, M. (1998): Enterprise Modelling. *AI Magazine,* Fall 1998, pp. 109-121. Menlo Park: AAAI Press.

[Frank, 1998]
Frank, U. (1998): Frank, U.: *The Memo Object Modelling Language (MEMO-OML).* Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 10. Universität Koblenz .

[Frank, 2002]
Frank, U. (2002) Multi-Perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In: *Proc. Hawaii Int. Conference on System Sciences (HICSS-35).*

[Frei et al., 1996]
H.-P. Frei, D. Harman, P. Schäuble & R. Wilkinson, eds. (1996): *Proc. 19th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*.

[Friedrich et al., 2002]
Friedrich, R., Iglezakis, I., Klein, W., and Pregizer, S. (2002): Experience-based Decision Support for Project Management with Case-Based Reasoning. In: *Proc. 1st German Workshop on Experience Management.* LNI Vol. P-10, pp. 139-150.

[Fuchs-Kittowski & Walter, 2002]
Fuchs-Kittowski, F.; Walter, R. (2002): Prozessorientierte Technikunterstützung für arbeitsprozessorientierte Weiterbildungen. In: M. Herczeg, H. Oberquelle & W. Prinz (eds.): *Mensch und Computer 2002 - Vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten. 2. Fachübergreifende Konferenz.* Stuttgart: Teubner.

[Fuhr, 1995]
Fuhr, N. (1995): Modelling Hypermedia Retrieval in Datalog. In: *Proc. HIM'95.* Konstanz: Universitätsverlag.

[Fünffinger et al., 2002]
Fünffinger, M., Rose, T., Ruprecht, C., Schott, H., Sieper, A. (2002): Management von Prozesswissen in projekthaften Prozessen. In: [Abecker et al., 2002c], pp. 293-320. *In German.*

[Gandon, 2002a]
Gandon, F. (2002): *Distributed Artificial Intelligence and Knowledge Management: Ontologies and Multi-Agent Systems for a Corporate Semantic Web.* Doctoral Thesis, Sophia-Antipolis: University of Nice.

[Gandon, 2002b]
Gandon, F. (2002): A Multi-Agent Architecture for Distributed Corporate Memories. In: *3rd Int. Symp. From Agent Theory to Agent Implementation at the*

16th European Meeting on Cybernetics and Systems Research (ECMSR 2002), pp. 623-628.

[Gandon, 2003]
Gandon, F. (2003): Agents Handling Annotation Distribution in a Corporate Semantic Web. *Web Intelligence and Agent Systems* **1**(1):23-45. Amsterdam: IOS Press.

[Gandon & Dieng-Kuntz, 2002]
Gandon, F. & Dieng-Kuntz, R. (2002): Distributed Artificial Intelligence for Distributed Corporate Knowledge Management. In: M. Klusch, S. Ossowski & O. Shehory, eds., *Cooperative Information Agents VI: 6th Int. Workshop CIA-2002.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2446.

[Gehle, 2001]
Gehle, M. (2001): IT-unterstützter Wissenstransfer in der internationalen Forschung & Entwicklung (Ein Praxisbericht des „Marktplatz des Wissens“ der BMW AG). In: *[Müller et al., 2001]. In German.*

[Georgakopoulos et al., 1995]
Georgakopoulos, D., Hornick, M. & Sheth, A. (1995): An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases* **3**(2):119-153. Boston: Kluwer Academic Publishers.

[Georgakopoulos & Tsalgatidou, 1998]
Georgakopoulos, D. & Tsalgatidou, A. (1998): Technology and Tools for Comprehensive Business Process Lifecycle Management. In: Dogac, A., Kalinichenko, L., Oszu, T. & Sheth, A. (eds.): *Workflow Management Systems and Interoperability.* NATO ASI Series F. Berlin, Heidelberg, New York: Springer-Verlag.

[Gery, 1991]
Gery, G. (1991): *Electronic Performance Support Systems: How and Why to Remake the Workplace Through the Strategic Application of Technology.* Boston / MA: Weingarten Publications.

[Gery, 2002]
Gery, G. (2002): Task Support, Reference, Instruction, or Collaboration? Factors in Determining Electronic Learning and Support Options. *Technical Communication* **49**(4): 420-427.

[Giunchiglia et al., 2002]
Giunchiglia, F., Mylopoulos, J. & Perini, A. (2002): The Tropos Software Development Methodology: Processes, Models and Diagrams. In: F. Giunchiglia, J. Odell & G. Weiß (eds.): *Agent-Oriented Software Engineering III, Third Int. Workshop, AOSE 2002.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2585, pp. 162-173.

[Goesmann et al., 1998]

Goesmann, T., Krämer, K., Striemer, R. & Wernsmann, C. (1998): Ein Kriterienkatalog zur Bestimmung der Eignung von Workflow-Management-Technologie zur Unterstützung von Geschäftsprozessen. In: Herrmann, T., Scheer, A.-W. & Weber, H. (eds.): *Verbesserung von Geschäftsprozessen mit flexiblen Workflow-Management-Systemen, Band 2.* Heidelberg. Pp. 96-105. *In German.*

[Goesmann & Herrmann, 2000
Goesmann, T. & Herrmann, T. (2000): Wissensmanagement und Geschäftsprozeßunterstützung am Beispiel des Workflow Memory Information System WoMIS. In: Th. Herrmann et al. (ed.s): *Verbesserung von Geschäftsprozessen mit flexiblen Workflow-Management-Systemen, Band 4.* Heidelberg: Physica-Verlag, 2000. *In German.*

[Goesmann, 2002]
Goesmann, T. (2002): *Ein Ansatz zur Unterstützung wissensintensiver Prozesse durch Workflow-Management Systeme.* Dissertation Technische Universität Berlin. *In German.*

[Gómez-Pérez, 1996]
Gómez-Pérez, A. (1996) : A Framework to Verify Knowledge Sharing Technology. *Expert Systems with Applications* **11**(4):519-529.

[Gordon & Domeshek, 1995]
Gordon, A.S. & Domeshek, E.A. (1995): Conceptual Indexing for Video Retrieval. In: M. Maybury, ed. (1995): *Intelligent Multimedia Information Retrieval, Working Notes of the IJCAI-95 Workshop, Montreal / Quebec*.

[Gräther & Prinz, 2001]
Gräther, W. & Prinz, W. (2001): The Social Web Cockpit: Support for Virtual Communities. In: *Proc. 2001 Int. ACM SIGGROUP Conference on Supporting Group Work.*

[Gronau & Laskowski, 2003]
Gronau, N., Laskowski, F. (2003): K_SERVICES: From State-of-the-Art Components to Next Generation Distributed KM Systems. In: Khosrow-Pour, M. (ed.): *Information Technology and Organizations: Trends, Issues, Challenges and Solutions.*

[Guarino et al., 1999]
Guarino, N., Masolo, C. & Vetere, G (1999): OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems,* **14**(3):70-80.

[Hagemeyer & Rolles, 1998]
Hagemeyer, J. & Rolles, R. (1998): Erhebung von Prozeßwissen für das Wissensmanagement. *Information Management & Consulting,* no. 13, pp. 46-50. *In German.*

[Hammer, 1990]
Hammer, M. (1990): Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review* July/August, pp. 104-112.

[Handschuh et al., 2003]
Handschuh, S., Staab, S. & Studer, R. (2003): Leveraging Metadata Creation for the Semantic Web with CREAM. In: A. Günter, R. Kruse & B. Neumann (eds.): *KI 2003: Advances in Artificial Intelligence, 26th Annual German Conference on AI.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2821, pp. 19-33.

[Handschuh & Staab, 2003]
Handschuh, S. & Staab, S., eds. (2003): *Annotation for the Semantic Web.* Amsterdam: IOS Press.

[Hansen et al., 1999]
Hansen, M.T., Nohria, N. & Tierney, T. (1999): What's Your Strategy for Managing Knowledge? *Harvard Business Review* **77**(2):106-116.

[Härtwig & Fähnrich, 2003]
Härtwig, J. & Fähnrich, K.P. (2003): Grundkonzepte des Wissensmanagements im Informationsraum. In: *Workshop Content- und Wissensmanagement der Leipziger Informatiktage 2003 (LIT'03). In German.*

[Hefke & Stojanovic, 2004]
Hefke, M. & Stojanovic, L. (2004): An Ontology-based Approach for Competence Bundling and Composition of ad-hoc Teams in an Organisation. In: I-KNOW-04. *To appear.*

[Hefke & Trunko, 2002]
Hefke, M. & Trunko, R. (2002): A Methodological Basis for Bringing Knowledge Management to Real-World Environments. In: [Karagiannis & Reimer, 2002], pp. 565-570.

[Heisig, 2001]
Heisig, P. (2001): Business-Process Oriented Knowledge Management. In: [Müller et al., 2001].

[Heijst et al., 1996]
Heijst, G. van, Spek, R. van der & Kruizinga, E. (1996): Organizing Corporate Memories. In: [Dieng & Vanwelkenhuisen, 1996].

[Heijst et al., 1997]
Heijst, G. van, Schreiber, T. & Wielinga, B. (1997): Using Explicit Ontologies in KBS Development. *Int. Journal on Human Computer Studies* **46**(2):183-292.

[Heijst et al., 1998]
G. van Heijst, R. van der Spek & E. Kruizinga (1998): The Lessons Learned Cycle. In: [Borghoff & Pareschi1998].

[Herterich et al., 2003]
Herterich, R., Houy, C., Müller, S. & Abecker, A. (2003): *The Process-Oriented Archive System of the DECOR Solution for Business-Process Oriented Knowledge Management.* Unpublished manuscript. URL: http://www.andreas-abecker.de/Downloads/Decor-DHC-2003.pdf [Last access: 04/20/2003]

[Hildreth & Kimble, 2002]
Hildreth, P.M. & Kimble, Ch. (2002): The Duality of Knowledge. *Information Research* **8**(1). URL: http://informationr.net/ir/8-1/paper142.html . [Last access: 02/14/2004].

[Hinkelmann et al., 2002]
Hinkelmann, K.; Karagiannis & D.; Telesko, R. (2002): PROMOTE – Methodologie und Werkzeug zum geschäftsprozessorientierten Wissensmanagement. In: [Abecker et al., 2002c]. *In German.*

[Hoof et al., 2003]
Hoof, A. van, Fillies, C. & Härtwig, J. (2003): Grundkonzepte des Wissensmanagements im Informationsraum. In: *Workshop Content- und Wissensmanagement der Leipziger Informatiktage 2003 (LIT'03). In German.*

[Huang, 1997]
Huang, K.-T. (1997): Capitalizing Collective Knowledge for Winning, Execution and Teamwork. Technical Whitepaper. URL: http://www.ibm.com/services/ articles/ intelcap.html [Last access: 03/31/1999].

[Huang, 1998]
Huang, K.-T. (1998): Capitalizing on Intellectual Assets. *IBM Systems Journal* 37(4): 570–583. URL: http://www.research.ibm.com/journal/sj/374/huang.html. [Last access: 05/29/2003]

[Hust, 2004]
Hust, A. (2004): Introducing Query Expansion Methods for Collaborative Information Retrieval. In: A. Dengel, M. Junker & A. Weisbecker (eds.): *Reading and Learning: Adaptive Content Recognition*. Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2956, pp. 252-280. *To appear.*

[Inkass, 2002]
Inkass Project Team (2002): *Survey of Knowledge Marketplaces.* Inkass Whitepaper. URL: http://www.inkass.com/library.htm [Last access: 04/24/2002]

[Inkass, 2003]
Inkass Project Team (2003): *Inkass Project Deliverable D8: Ontologies and Knowledge Asset Descriptions.* Internal Report.

[Jablonski et al., 2002]
Jablonski, S., Horn, S. & Schlundt, M. (2002): Prozessorientierte Wissensmanagement mit der i>WorkBench. In: [Abecker et al., 2002c], pp. 249-274. *In German.*

[JDBC, 1998]
JDBC Homepage at SUN. http://java.sun.com/products/jdbc/ [Last access: 03/31/2004]

[Jaeger & Prakash, 1995]

Jaeger, T. & Prakash, A. (1995): Management and Utilization of Knowledge for the Automatic Improvement of Workflow Performance. In: *COCS'95 – Proc. Conference on Organizational Computing Systems.* New York: ACM Press.

[Jennings et al., 1996]
Jennings, N.R., Faratin, P., Johnson, M.J., Norman, T.J., O'Brien, P., & Wiegand, M.E. (1996): Agent-Based Business Process Management. *Int. Journal on Cooperative Information Systems* **5**(2&3):105-130.

[Jensen & Nilsson, 2003]
Jensen, P.A. & Nilsson, J.F. (2003): Ontology-Based Semantics for Prepositions. In: *ACL-SIGSEM workshop: The Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications* . URL: http://www.humaniora.sdu.dk/~paj/toulouse2.pdf [Last acces: 04/18/2004]

[Joeris et al., 1997]
Joeris, G., Klauck, C. & Herzog, O. (1997): Dynamical and Distributed Process Management Based on Agent Technology. In: G. Grahne (ed.): *SCAI'97 – Proc. Sixth Scandinavian Conference on Artificial Intelligence.* Amsterdam: IOS Press. Frontiers in Artificial Intelligence and Applications, Vol. 40, pp. 187-198.

[Junginger et al., 2000]
Junginger, S., Kühn, H., Strobl, R. & Karagiannis, D. (2000): Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation – ADONIS: Konzeption und Anwendungen. *Wirtschaftsinformatik* **42**(5): 392-401. *In German.*

[Junker & Hoch, 1998]
Junker, M. & Hoch, R. (1998): An Experimental Evaluation of OCR Text Representations for Learning Document Classifiers. *Int. Journal on Document Analysis and Recognition*, **1**(2):116-122.

[Junker, 2001]
Junker, M. (2001): *Lernen heuristischer Regeln für die Textkategorisierung.* Ph.D. thesis, University of Kaiserslautern / Germany. *In German.*

[Kafentzis et al., 2002]
Kafentzis, K., Apostolou, D., Mentzas, G. & Georgolios, P. (2002): A Framework for Analysis and a Review of Knowledge Asset Marketplaces. In: [Karagiannis & Reimer, 2002], pp. 301-313.

[Kamiya et al., 1996]
Kamiya, K., Roscheisen, M. & Winograd, T. (1996): Grassroots: Providing a Uniform Framework for Communicating, Sharing Information, and Organizing People. *ACM CHI-96 Conference on Human Factors in Computing Systems*.

[KAON, 2004]
KAON - The Karlsruhe Ontology and Semantic Web Framework. Developer's Guide for KAON 1.2.7. Version January 2004. URL: http://kaon.semanticweb.org/Members/rvo/KAON_Dev_Guide.pdf . [Last access: 03/10/2004]

[Karagiannis et al., 1996]
Karagiannis, D., Junginger, S. & Strobl, R. (1996): Introduction to Business Process Management Systems Concepts. In: B. Scholz-Reuter and E. Stickel, editors, *Business Process Management*, Lecture Notes on Computer Science, LNCS. Berlin, Heidelberg, London, New York: Springer-Verlag.

[Karagiannis & Reimer, 2002]
Karagiannis, D. & Reimer, U., eds. (2002): *Practical Aspects of Knowledge Management, 4th Int. Conference, PAKM 2002.* Berlin, Heidelberg, New York: Springer-Verlag, LNCS 2569.

[Kindermann et al., 1996]
Kindermann, C. et al. (1996)*: The MIHMA Demonstrator Application: bmt line.* Technical report, Non Standard Logics S.A., Paris and TU Berlin.

[Kiss & Quinqueton, 2001]
Kiss, A. & Quinqueton, J. (2001): Learning User Preferences in Multi-agent System. In: B. Dunin-Keplicz & E. Nawarecki (eds.): *From Theory to Practice in Multi-Agent Systems, 2nd Int. WS of Central and Eastern Europe on Multi-Agent Systems, CEEMAS 2001.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2296, pp. 169-178.

[Klamma et al., 1998]
Klamma, R., Peters, P. & Jarke, M. (1998): Workflow Support for Failure Management in Federated Organizations. In: Blanning, R.W. & King, D.R. (eds.): *Proc. 31st Hawaii Int. Conference on System Sciences (HICCS-31),* pp. 302-311. IEEE Computer Society Press.

[Klamma & Jarke, 1999]
Klamma, R. & Jarke, M. (1999): Knowledge Management Cultures: A Comparison of Engineering and Cultural Science projects. In: *ECSCW-Workshop XMWS'99, Beyond Knowledge Management: Managing Expertise,* Kopenhagen / Denmark. URL: http://www.cs.uni-bonn.de/~prosec/ECSCW-XMWS/ [Last access: 04/18/2004]

[Klamma, 2000]
Klamma, R. (2000): *Vernetztes Verbesserungsmanagement mit einem Unternehmensgedächtnis-Repository.* Dissertation, RWTH Aachen.

[Klamma & Schlaphof, 2000]
Klamma, R. & Schaphof, S. (2000): Rapid Knowledge Deployment in an Organizational-Memory-Based Workflow Environment. In: *Proc. of the 8th European Conference on Information Systems (ECIS 2000).* Pp. 364-371.

[Klemke, 2002]
Klemke, R. (2002): *Modelling Context in Information Brokering Processes. Dissertation*. RWTH Aachen.

[Klein & Abecker, 1999]

Klein, B. & Abecker, A. (1999): Towards Distributed Knowledge-based Parsing for Document Analysis and Understanding. In: *Proc. IEEE Forum on Research and Technology Advances in Digital Libraries, Sixth IEEE ADL '99.* Los Alamitos / CA: IEEE Computer Society.

[Klein & Dengel, 2004]
Klein, B. & Dengel, A.R. (2004): Problem-Adaptable Document Analysis and Understanding for High-Volume Applications. *International Journal on Document Analysis and Recognition* **6**(3):167-180. Heidelberg: Springer-Verlag.

[Kleinhans, 1989]
Kleinhans, A.M. (1989): *Wissensverarbeitung im Management.* Frankfurt am Main: Verlag Peter Lang. *In German.*

[Klusch et al., 2003]
Klusch, M., Ossowski, S., Omicini, A. & Laamanen, H., eds. (2003): *Cooperative Information Agents VII, 7th International Workshop, CIA-2003.* Berlin, Heidelberg, New York: Springer-Verlag.

[Klusch et al., 2002]
Klusch, M., Ossowski, S. & Shehory, O., eds. (2002): *Cooperative Information Agents VI, 6th International Workshop, CIA- 2002.* Berlin, Heidelberg, New York: Springer-Verlag.

[Knapp, 1998]
Knapp, E. (1998): Knowledge Management: The Key to Success in the 21st Century. *European Business Information Conference 1998.*

[Know-Net, 1999]
Know-Net (1999): *Deliverable 1.1: Architectural Framework and Holistic Requirements for Know-Net*. Internal Project Documentation. Can be obtained from Project Coordinator.March 1999.

[Kobsa, 2001]
Kobsa, A. (2001): Generic User Modeling Systems. *User Modeling and User-Adapted Interaction* **11**(1-2):49-63.

[Kühn & Abecker, 1997]
Kühn, O. & Abecker, A. (1997): Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges. *Journal of Universal Computer Science*. **3**(8), Special Issue on Information Technology for Knowledge Management: Berlin: Springer Science Online. Also in: [Borghoff & Pareschi, 1998].

[Kühn & Höfling, 1994]
Kühn, O. & Höfling, B. (1994): Conserving Corporate Knowledge for Crankshaft Design. In*: IEA/AIE´94 - Proceedings of the Seventh Int. Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems.* Pp. 475-484. ACM Press.

[Kushmerick, 2000]
Kushmerick, N. (2000): Wrapper Induction: Efficiency and Expressiveness.

Artifical Intelligence **118**(1-2):15-68. Essex: Elsevier Science Publishers Ltd.

[Laffey, 1995]
Laffey, J. (1995): Dynamism in Electronic Performance Support Systems. *Performance Improvement Quarterly* **8**(1):31-46.

[Lattner & Apitz, 2002]
Lattner, A.D. & Apitz, R. (2002): A Metadata Generation Framework for Heterogeneous Information Sources. In: *2nd International Conference on Knowledge Management (I-KNOW '02),* pp. 164-169.

[Lattner & Herzog, 2003]
Lattner, A.D. & Herzog, O. (2003): Instance-Based Learning and Information Extraction for the Generation of Metadata. In: *3rd International Conference on Knowledge Management (I-KNOW '03),* pp. 472-479.

[Lehner, 2000]
Lehner, F. (2000): *Organisational Memory.* München: Carl Hanser Verlag. *In German.*

[Lehner & Maier, 2000]
Lehner, F. & Maier, R. (2000): How Can Organizational Memory Theories Contribute to Organizational Memory Systems? In: [Abecker et al., 2000d].

[Leighton, 2004]
Leighton, C. (2004): *What is an EPSS?*

URL: http://it2.coe.uga.edu/EPSS/Whatis.html [Last access: 04/03/2004].

[Leonard-Barton, 1995]
Leonard-Barton, D. (1995): *The Wellsprings of Knowledge – Building and Sustaining the Source of Innovation.* Harvard Business School Press.

[Liebowitz & Wilcox, 1997]
Liebowitz, J. & Wilcox, L.C., eds. (1997): *Knowledge Management and Its Integrative Elements.* Boca Raton: CRC Press.

[Liao et al., 1999]
Liao, M., Hinkelmann, K., Abecker, A. & Sintek, M. (1999): A Competence Knowledge Base System as Part of the Organizational Memory. In: *German National Conference on Knowledge-Based Systems (XPS-99)*, number 1570 in LNAI. Berlin, Heidelberg, New York: Springer-Verlag.

[Liao et al., 1999b]
Liao, M., Abecker, A., Bernardi, A., Hinkelmann, K. & Sintek, M. (1999): Ontologies for Knowledge Retrieval in Organizational Memories. In: *Workshop on Learning Software Organizations (LSO)* at SEKE'99, Kaiserslautern / Germany.

[Lotus, 2003]
Lotus (2003): *The Lotus Knowledge Management Framework. Technology White-*

paper. URL: http://www.kolaco.com. [Last access: 05/29/2003].

[Maass et al., 2003]
Maass, W., Eickhoff, W.-C., Stahl, F. & Schaefer, M.-F. (2003): *A Generic Structure for Information Objects in Knowledge Sharing and Trading Contexts.* Mcminstitute-Working Paper. St. Gallen: University of St. Gallen.

[Mahé & Rieu, 1998]
Mahé, S. & Rieu, C. (1998): A Pull Approach to Knowledge Management: Using IS as a Knowledge Indicator to Help People Know when to Look for Knowledge Reuse. In: U. Reimer (ed.): *PAKM 98, Practical Aspects of Knowledge Management.* CEUR Proceedings, Vol. 13.

[Maedche, 2002]
Maedche, A. (2002): *Ontology Learning for the Semantic Web.* Boston: Kluwer Academic Publishers.

[Maedche et al., 2001]
Maedche, A., Staab, S., Stojanovic, N., Studer, R. & Sure, Y. (2001): SEAL – A Framework for Developing Semantic Web PortALs. In: B. Read (ed.): *Advances in Databases: 18th British National Conference on Databases, BNCOD 18 Chilton, UK.* LNCS 2097, pp. 1-22. Berlin, Heidelberg, New York: Springer-Verlag.

[Maedche et al., 2003]
Maedche, A., Motik, B. & Stojanovic, L. (2003): Managing Multiple and Distributed Ontologies on the Semantic Web. *VLDB Journal* **12**(4): 286-302.

[Maedche & Staab, 2002]
Maedche, A. & Staab, S. (2002): Measuring Similarity between Ontologies. In: A. Gómez-Pérez & V.R. Benjamins (eds.): *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th Int. Conf. EKAW 2002.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2473, pp. 251-263.

[Maier, 2004]
Maier, R. (2004): *Knowledge Management Systems: Information and Communication Technologies for Knowledge Management, 2nd edition.* Berlin, Heidelberg, New York: Springer-Verlag.

[Malhotra, 2001]
Malhotra, Y. (2001): Expert Systems for Knowledge Management: Crossing the Chasm between Information Processing and Sense Making. *Expert Systems with Applications*, 20(2001):7-16. Amsterdam: Elsevier Science Ltd.

[Malone et al., 1999]
Malone, T.W. et al. (1999): Tools for Inventing Organizations: Toward a Handbook of Organizational Processes. *Management Science* **45**(3):425-443. URL: http://ccs.mit.edu/dell/papers/ph-mgtsci1999.pdf . [Last access: 03/31/2004].

[Malone et al., 2003]
Malone, T.W., Crowston, K. & Herman, G.A., eds. (2003): *Organizing Business Knowledge: The MIT Process Handbook.* Cambridge / MA: MIT Press.

[Martens, 2003]
Martens, A. (2003): *Folien zur Vorlesung "Geschäftsprozessmodellierung" an der Humboldt-Universität zu Berlin,* WS 2002/03. *In German.*

[Matsuda, 1993]
Matsuda, T. (1993): Organizational Intelligence als Prozeß und als Produkt. *technologie & management* **42**(1):12-17. *In German.*

[Maurer, 1999]
Maurer, H. (1999): The Heart of the Problem: Knowledge Management and Knowledge Transfer. In: *Enable'99*, Espoo-Vanta Institute of Technology, pp. 8–17.

[Maus, 2001]
Maus, H. (2001): Workflow Context as a Means for Intelligent Information Support. In: V. Akman, P. Bouquet, R. Thomason & R.A. Young (eds.): Modeling and Using Context. Third International and Interdisciplinary Conference CONTEXT-2001, Dundee / UK. LNAI 2116. Berlin, Heidelberg, New York: Springer-Verlag.

[McDermott, 2000]
McDermott, R. (2000): *Knowing in Community: 10 Critical Success Factors in Building Communities of Practice.* Online publication: http://www.co-i-l.com/coil/knowledge-garden/cop/knowing.shtml . [Last access: 02/15/2004].

[McDonald, 2001]
McDonald, D.W. (2001): Evaluating Expertise Recommendations. In: *Proc. Of the 2001 Int. ACM SIGGROUP Conference on Supporting Group Work.*

[McGuinness, 1998]
McGuinness, D.L. (1998): Ontological Issues for Knowledge-Enhanced Search. In N. Guarino, ed.: *Int'l Conference on Formal Ontology in Information Systems (FOIS'98)*. Amsterdam: IOS Press.

[Mecella et al., 2002]
Mecella, M., Scannapieco, M., Virgillito, A., Baldoni, R., Catarci, T. & Batini, C. (2002): Managing Data Quality in Cooperative Information Systems. In: [Meersman & Tari, 2002], pp. 486-502.

[Meersman & Tari, 2002]
Meersman, R. & Tari, Z., eds. (2002): *On the Move to Meaningful Internet Systems - Confederated Int. Conferences DOA, CoopIS and ODBASE 2002.* LNCS 2519. Berlin, Heidelberg, New York: Springer-Verlag.

[Jarrar & Meersman, 2002]

Jarrar, R. & Meersman, R.(2002): Formal Ontology Engineering in the DOGMA Approach. In: Meersman, R. et al. (eds.): *Proc. Confederated Int. Conferences: On the Move to Meaningful Internet Systems (CoopIS, DOA, ODBASE 2002).* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2519, pp. 1238-1254.

[Malhotra, 2003]
Malhotra, Y. (2003): Why Knowledge Management Systems Fail: Enablers and Constraints of Knowledge Management in Human Enterprises. In: Holsapple, C.W., *Handbook on Knowledge Management 2 – Knowledge Directions,* pp. 577-599. Berlin, Heidelberg, New York: Springer-Verlag.

[Meghini et al., 1991]
Meghini, C., Rabitti, F. & Thanos, C. (1991): Conceptual Modeling of Multimedia Documents. *IEEE Computer* **24**(10):23-30. Los Alamitos: IEEE Computer Society Press.

[Meghini et al., 2001]
Meghini, C., Sebastiani, F. & Straccia, U. (2001): A Model of Multimedia Information Retrieval. *Journal of the ACM (JACM)* **48**(5):909-970. New York: ACM Press.

[Meghini & Straccia, 1996]
Meghini, C. & Straccia, U. (1996): A Relevance Terminological Logic for Information Retrieval. In: [Frei et al., 1996].

[Mentzas, 1999]
Mentzas, G. (1999): Coupling Object-Oriented and Workflow Modelling in Business and Information Process Reengineering. *Information Knowledge Systems Management* **1**(1):63-87. Amsterdam: IOS Press.

[Mentzas et al., 2002]
Mentzas, G., Apostolou, D., Young, R. & Abecker, A. (2002): *Knowledge Asset Management – Beyond the Product-centric and the Process-centric Approach.* Berlin, New York, Heidelberg: Springer-Verlag.

[Mentzas et al., 2001]
Mentzas, G., Apostolou, D., Young, R. & Abecker, A. (2001): Knowledge Networking: A Holistic Approach, Method and Tool for Leveraging Corporate Knowledge. *Journal of Knowledge Management* **5**(1):94-106, MCB University Press.

[Mertins et al., 2003a]
Mertins, K., Heisig, P. & Alwert, K. (2003): Process-Oriented Knowledge Structuring. *Journal of Universal Computer Science* **9**(6):542-550.

[Mertins et al., 2003b]
Mertins, K., Heisig, P. & Vorbeck, J., eds. (2003): *Knowledge Management. Concepts and Best Practices. 2nd edition.* Berlin, Heidelberg, New York : Springer-Verlag.

[Michelis et al., 1996]
Michelis, G. de, Dubois, E., Jarke, M., Matthes, F., Mylopoulos, J., Pohl, K., Schmidt, J., Woo, C. & Yu, E. (1996) : *Cooperative Information Systems – A Manifesto.* URL : http://www.sts.tu-harburg.de/papers/1997/DDJ+97.pdf [Last access : 04/18/2004]

[Michelson, 2001]
Michelson, M. (2001): *Betriebliche Informationsorganisation II.* Vorlesungsskript HBI Stuttgart.

[Morrison, 1997]
Morrison, J. (1997): Organizational Memory Information Systems: Characteristics and Development Strategies. In: *Proc. 30th Annual Hawaii Int. Conference on System Sciences (HICSS-30).* Los Alamitos / CA: IEEE Computer Society Press.

[Muehlen, 1999]
Muehlen, M. zur (1999): Evaluation of Workflow Management Systems Using Meta Models. In: Sprague, R.Jr. (ed.): *Proc. 32nd Annual Hawaii Int. Conference on Systems Sciences.*

[Mulholland et al., 2001]
Mulholland, P., Zdrahal, Z., Domingue, J. Hatala, M. & Bernardi, A. (2001): *A Methodological Approach to Supporting Organizational Learning.* International Journal of Human-Computer Studies, **55**(3):337-367.

[Müller, 2002]
Müller, M.E. (2002): *Inducing Conceptual User Models.* Ph.D. Thesis, Institute of Cognitive Science, University of Osnabrück.

[Müller et al., 2001]
Müller, H.-J., Abecker, A., Maus, H. & Hinkelmann, K., eds. (2001): *Workshop „Geschäftsprozessorientiertes Wissensmanagement – Von der Strategie zum Content"* as part of WM-2001, Baden-Baden. CEUR-Volume 37 Online. URL:
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-37/ .
[Last access: 03/31/2004]. *In German.*

[Müller et al., 2004]
Müller, H.-J., Telschow, A., Xia, W. & Aydin, A. (2004): Wissenszentrierte Prozessanalyse. Whitepaper. Ludwigshafen: BeeBetter GmbH.
URL: http://www.beebetter.de/themen/wm_mittelstand/text.html
[Last access: 04/18/2004]

[Müller & Herterich, 2001]
Müller, S. & Herterich, R. (2001): Prozessorientiertes Wissensmanagement mit CognoVision. In [Müller et al., 2001]. *In German.*

[Muslea et al., 2003]
Muslea, I., Minton, S. & Knoblock, C. (2003): Active Learning With Strong and Weak Views: A Case Study on Wrapper Induction. In: *Proc. of the Int. Joint*

Conference on Artificial Intelligence (IJCAI 2003).

[Muslea, 2002]
Muslea, I. (2002): *Active Learning with Multiple Views.* Ph.D. dissertation at the University of Southern California.

[Myers & Berry, 1999]
Myers, K.L. & Berry, P.M. (1999): *Workflow Management Systems: An AI Perspective.* Technical Report, Artificial Intelligence Center, SRI International, Menlo Park / CA: SRI International.

[Mylopoulos, 2003]
Mylopoulos, J. (2003): Ten Years of CoopIS Conferences: Cooperative Information Systems Then and Now. Keynote speech at the *11th IFCIS Conference on Cooperative Information Systems (CoopIS'03),* Catania / Italy. URL:
http://www.cs.rmit.edu.au/fedconf/2003/keynotes/OTM03-John.pdf
[Last access: 04/13/2004]

[Nabeth et al., 2003]
Nabeth, T., Angehrn, A. & C. Roda (2003): Enhancing Knowledge Management Systems with Cognitive Agents. *Systèmes d'Information & Management* **8**(2).

[Nabeth et al., 2004]
Nabeth, T., Angehrn, A. & Balakrishnan, R. (2004): Integrating 'Context' in e-Learning Systems Design. *IEEE International Conference on Advanced Learning Technologies (ICALT 2004),* Joensuu, Finland, submitted.

[Nakata et al., 1998]
Nakata, K., Voss, A., Juhnke, M. & Kreifelts, T. (1998): Collaborative Concept Extraction from Documents. In U. Reimer (ed.): *Proc. 2nd Int. Conference on Practical Aspects of Knowledge Management (PAKM 98).*

[Nonaka, 1991]
Nonaka, I. (1991): The Knowledge-Creating Company. *Harvard Business Review,* 69, November-December, pp. 96-104.

[Nonaka & Takeuchi, 1995]
Nonaka, I. & Takeuchi, H. (1995): *The knowledge-creating company : How japanese companies create the dynamics of innovation*. Oxford University Press.

[North, 1999]
North, K. (1999): *Wissensorientierte Unternehmensführung – Wertschöpfung durch Wissen.* Wiesbaden: Gabler. *In German.*

[Nüttgens & Rump, 2002]
Nüttgens, M. & Rump, F.J. (2002): Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: *Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen (Promise'2002),* Potsdam. *In German.*

[Oberle et al., 2003]
Oberle, D., Berendt, B., Hotho, A., & Gonzalez, J. (2003): Conceptual User Tracking. In: E.M. Ruiz, J. Segovia, & P.S. Szczepaniak (eds.), Web Intelligence, First Int. Atlantic Web Intelligence Conference, AWIC 2003. Berlin, Heidelberg, New York: Springer-Verlag. LNCS 2663, pp. 155-164.

[Oberschulte, 1994]
Oberschulte, H. (1994): *Organisatorische Intelligenz : ein integrativer Ansatz des organisatorischen Lernens.* Dissertation an der Universität des Saarlandes, Saarbrücken. München [u.a.]: Hampp*. In German.*

[Oberweis et al., 1994]
Oberweis, A., Scherrer, G. & Stucky, W. (1994): INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems. *Information Systems* **19**(8):643-682.

[Oberweis et al., 2001]
Oberweis, A., Paulzen, O. & Sexauer, H.J. (2001): Ein wissensbasiertes Vorgehensmodell zur Gestaltung von CRM-Systemen. In: Bauknecht, K., Brauer, W. & Mück, T.A. (eds.): *Informatik 2001. Wirtschaft und Wissenschaft in der Network Economy – Visionen und Wirklichkeit.* Pp. 429-436. *In German.*

[Oberweis & Paulzen, 2003]
Oberweis, A. & Paulzen, O. (2003): Kontinuierliche Qualitätsverbesserung im Wissensmanagement - ein prozessbasiertes Reifegradmodell. In: *Proc. KnowTech 2003: 5. Konferenz zum Einsatz von Knowledge Management in Wirtschaft und Verwaltung.*

[Oestereich et al., 2003]
Oestereich, B., Weiss, C., Schröder, C., Weilkiens, T. & Lenhard, A. (2003): *Objektorientierte Geschäftsprozessmodellierung mit der UML.* Dpunkt Verlag. *In German.*

[O'Leary, 1998]
O'Leary, D. (1998): Using AI in Knowledge Management: Knowledge Bases and Ontologies. *IEEE Intelligent Systems*, May/June 1998.

[O'Leary & Selfridge, 2000]
O'Leary, D. & Selfridge, P. (2000): Knowledge Management for Best Practices. *Communications of the ACM* **43**(11es):281-291.

[Ontoprise, 2003]
Ontoprise GmbH (2003): *OntoOffice Tutorial.* Karlsruhe: Ontoprise GmbH. URL: http://www.ontoprise.de/customercenter/support/tutorials Last access: 04/11/2004.

[Ovum, 1998]
Ovum (1998): *Knowledge management: Applications, markets, and technologies*. Ovum Ltd.

[Palkovits et al., 2003]

Palkovits, S., Woitsch, R. & Karagiannis, D.: Process-Based Knowledge Management and Modelling in E-government – An Inevitable Combination. In: *KMGov 2003.* Pp. 213-218.

[Papavassiliou et al., 2002]
G. Papavassiliou, G. Mentzas & A. Abecker (2002): Integrating Knowledge Modelling in Business Process Management. In: *The Xth European Conference on Information Systems (ECIS-2002).*

[Papavassiliou et al., 2002b]
G. Papavassiliou, S. Ntioudis, G. Mentzas & A. Abecker (2002): Managing Knowledge in Weakly-structured Administrative Processes. Online Proceedings *Third European Conference on Organizational Knowledge, Learning, and Capabilities (OKLC-2002)*, Athens / Greece.

[Papavassiliou et al., 2002c]
G. Papavassiliou, S. Ntioudis, G. Mentzas & A. Abecker (2002): Business Process Knowledge Modelling: Method and Tool. In: *Third Int. Workshop on Theory and Applications of Knowledge Management (TAKMA 2002).* In Conjunction with DEXA 2002, Aix-en-Provence / France.

[Papavassiliou et al., 2003]
G. Papavassiliou, S. Ntioudis, G. Mentzas & A. Abecker (2003): Business Process Modelling and Enactment for Task-Specific Information Support. In: W. Uhr, W. Esswein & E. Schoop (eds.): *WI-2003, 6. Int. Tagung Wirtschaftsinformatik 2003: Medien - Märkte - Mobilität / Band I*, pp. 977-996. Heidelberg: Physica Verlag.

[Papavassiliou et al., 2003b]
G. Papavassiliou, S. Ntioudis, G. Mentzas & A. Abecker (2003): Supporting Knowledge-Intensive Work in Public Administration Processes. *Knowledge and Process Management:The Journal of Corporate Transformation* **10**(3):164-174. Hoboken: John Wiley & Sons.

[Partridge & Stefanova, 2001]
Partridge, C. & M. Stefanova (2001): A Synthesis of State of the Art Enterprise Ontologies. Padova / Italy: The BORO Program, LADSEB CNR. URL: http://www.boroprogram.org/pdfs_trap/OES-01.pdf. [Last access: 04/06/2004]

[Partridge, 2002]
Partridge, C. (2002): *The CEO Project: An Introduction.* Technical Report 07/02. Padova / Italy: LADSEB-CNR. URL: http://www.boroprogram.dsl.pipex.com/ladsebreports/ladseb_t_r_07-02.pdf. [Last access: 04/06/2004]

[Parunak, 1998]
Parunak, H. van Dyke (1998): What Can Agents Do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC. In: M. Klusch & G. Weiß (eds.): *Cooperative Information Agents II, Learning, Mobility and Electronic Commerce for Information Discovery on the Internet.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 1435, pp. 1-18.

[Paulzen & Haas, 2002]
Paulzen, O. & Haas, S. (2002): Integration von Knowlege Warehouses und Knowlege Networks – Konzept und Methodik am Beispiel des Intelligent Supplier Management. In: Herget, J. (ed.): *Competitive & Business Intelligence – Neue Konzepte, Methoden & Instrumente,* Konstanz/Berlin 2002, pp. 29-50. *In German.*

[Paulzen & Perz, 2000]
Paulzen, O. & Perc, P. (2000): A Maturity Model for Quality Improvement in Knowledge Management. In: Wenn, A.; McGrath, M. & Burstein, F. (eds.): *Enabling Organisations and Society through Information Systems, Proc. 13th Australasian Conference on Information Systems (ACIS 2002),* pp. 243-253.

[Perich, 2004]
Perich, F. (2004): *MoGATU BDI Ontology.* URL: http://mogatu.umbc.edu/bdi/ Last access: 04/12/2004

[Picot & Rohrbach, 1995]
Picot, A. & Rohrbach, P. (1995): Organisatorische Aspekte von Workflow-Management-Systemen. *Information Management,* Nr. 1.

[Polanyi, 1966]
Polanyi, M. (1966). *The Tacit Dimension.* London: Routledge & Kegan Paul Ltd.

[Polterauer & Mayrhofer, 1999]
Polterauer, A. & Mayrhofer, D. (1999): Practical Applications of Knowledge Management – Reference Models for Key Issues. In: *ECSCW-Workshop XMWS'99, Beyond Knowledge Management: Managing Expertise,* Kopenhagen / Denmark. URL: http://www.cs.uni-bonn.de/~prosec/ECSCW-XMWS/ [Last access: 04/18/2004]

[Power, 2002]
Power, D.J. (2002): *Decision Support Systems: Concepts and Resources for Managers.* Westport: Greenwood Publishing Group.

[Probst et al., 1999]
Probst, G., Raub, S. & Romhardt, K. (1999): *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen (3. Auflage)*, Wiesbaden: Gabler. *In German.*

[Quesenbery, 2002]
Quesenbery, W. (2002): Who is in Control? The Logic Underlying the Intelligent Technologies Used in Performance Support. *Technical Communication* **49**(4):449-457.

[Qiu & Riesbeck, 2004]
Qiu, L. & Riesbeck, C.K. (2004): An Incremental Model for Developing Educational Critiquing Systems.In *Proc. World Conference on Educational Multi-media, Hypermedia & Telecommunications (ED-MEDIA).*

[Ramesh, 1997]
Ramesh, B. (1997): Towards a Meta-Model for Representing Organizational Memory. In: *Proc. 30th Annual Hawaii Int. Conference on System Sciences (HICSS-30).* Los Alamitos / CA: IEEE Computer Society Press.

[Rao & Goldmann-Segall, 1995]
Rao, V. &Goldmann-Segall, R. (1995): Capturing Stories in Organizational Memory Systems: The Role of Multimedia. In: Sprague, R. & Nunamaker, J. (eds.): *Proc. 28th Annual Hawaii Int. Conf. on System Sciences, Vol. III.* Pp. 333-341.

[Rehäuser & Krcmar, 1996]
Rehäuser, Jacob. & Krcmar, H. (1996): Wissensmanagement im Unternehmen. In: Schreyögg, G. & Conrad, P. (eds.): *Managementforschung 6.* Wissensmanagement. Berlin, New York. S. 5-40. *In German.*

[Reimer et al., 1998]
Reimer, U., Margelisch, A., Novotny, B. & Vetterli, T. (1998): EULE2: A Knowledge-Based System for Supporting Office Work. *ACM SIGGROUP Bulletin*, **19**(1):56-61.

[Reimer et al., 2000]
Reimer, U., Margelisch A., & Staudt, M. (2000): A Knowledge-Based Approach to Support Business Processes. In: *Bringing Knowledge to Business Processes, Workshop in the AAAI Spring Symposium Series 2000, Stanford, CA.*

[Reinmann-Rothmeier & Mandl, 2000]
Reinmann-Rothmeier, G. & Mandl, H. (2000): *Individuelles Wissensmanagement. Strategien für den persönlichen Umgang mit Wissen am Arbeitsplatz.* Bern: Huber. *In German.*

[Remus, 2002]
Remus, U. (2002): *Prozeßorientiertes Wissensmanagement. Konzepte und Modellierung.* Ph.D. Thesis, University of Regensburg. *In German.*

[Remus, 2002b]
Remus, U. (2002): Integrierte Prozess- und Kommunikationsmodellierung zur Verbesserung von wissensintensiven Geschäftsprozessen. In: [Abecker et al., 2002c], pp. 91-122. *In German.*

[Richter, 1995]
M.M. Richter (1995): *The Knowledge Contained in Similarity Measures.* Invited talk at ICCBR95.

[Rijsbergen, 1989]
C.J. van Rijsbergen (1989): Towards an Information Logic. In: *Proc. of the 12th Annual Int. ACM SIGIR Conf. On Research and Development in Information Retrieval.*

[Rist et al., 2003]
T. Rist, E. André & S. Baldes (2003): A Flexible Platform for Building Applications with Life-Like Characters. In: *Proc. of the 2003 Int. Conference on*

Intelligent User Interfaces, pp. 158-165. ACM.

[Rocchio, 1971]
J.J. Rocchio (1971): Relevance Feedback in Information Retrieval. In G. Salton (ed.): *The SMART Retrieval System - Experiments in Automatic Document Processing.* Englewood Cliffs / NJ: Prentice Hall.

[Rölleke & Fuhr,1996]
Th. Rölleke & N. Fuhr (1996): Retrieval of Complex Objects Using a Four-Valued Logic. In: [Frei et al., 1996].

[Rombach, 1988]
H.D. Rombach (1988): A Specification Framework for Software Processes: Formal Specification and Derivation of Information Base Requirements. In: C. Tully (ed.): *Proc. Fourth Int. Software Process Workshop (ISPW '88).* IEEE Computer Society Press. Pp. 142-147.

[Romhardt & Probst, 1997]
Romhardt, K. & Probst, G. (1997): Building Blocks of Knowledge Management – A Practical Approach. Paper presented at the seminar *"Knowledge Management and the European Union"*, Utrecht / Netherlands.

[Rose et al., 2002]
T. Rose, M. Fünffinger, H. Knublauch & C. Rupprecht (2002): Prozessorientiertes Wissensmanagement. KI – *Künstliche Intelligenz* 1/2004, pp.19-24. *In German*.

[Roth-Berghofer & Iglezakis, 2000]
Roth-Berghofer, Th. & Iglezakis, I. (2000): Developing an Integrated Multilevel Help-Desk Support System. In: *Proc. 8th German Workshop on Case—Based Reasoning*, pp 145-155. Ulm: DaimlerChrysler, Research and Technology, FT3/KL.

[Rupprecht, 2002]
Rupprecht, Christian (2002): *Ein Konzept zur projektspezifischen Individualisierung von Prozessmodellen.* Dissertation an der Universität Karlsruhe, Fak. f. Wirtschaftswissenschaften. *In German.*

[Scacchi & Valente, 1999]
Scacchi, W. & Valente, A. (1999): *Developing a Knowledge Web for Business Process Redesign.* URL: http://www.ics.uci.edu/~wsacchi/Papers/KnowledgeWeb/ . Last access: 03/30/2004.

[Scheer, 2001]
Scheer, A.-W. (2001): *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen, 4. Aufl.* Berlin, Heidelberg, New York: Springer-Verlag. *In German.*

[Scheir, 2002]
Scheir, P. (2002): *Supporting Customer Relationship Management through Knowledge Management.* Master's Thesis. Institute for Information Systems and Computer Media IICM, TU Graz.

[Schmalhofer & Elst, 1999]
Schmalhofer, F. & Elst, L. van (1999): An Oligo-Agents System with Shared Responsibilities for Knowledge Management. In: D. Fensel & R. Studer (eds.), *Knowledge Acquisition, Modeling and Management, Proc. 11th European Workshop, EKAW '99,* pp. 379-384. Berlin, Heidelberg, New York: Springer-Verlag.

[Schmaltz & Hagenhoff, 2003]
Schmaltz, R. & Hagenhoff, S. (2003): *Entwicklung von Anwendungssystemen für das Wissensmanagement: State of the Art der Literatur.* Arbeitsbericht Nr. 5/2003. Universität Göttingen: Institut für Wirtschaftsinformatik. *In German.*

[Schmid et al., 1999]
Schmid, R.E., Kaiser, T.M., Bach, V. & Österle, H. (1999): A Process-oriented Framework for Efficient Intranet Management. In: INET'99. URL: http://www.isoc.org/inet99/proceedings/1d/1d_4.htm [Last access: 04/15/2004]

[Schmiedel & Volle, 1996]
A. Schmiedel & Ph. Volle (1996): Using Structured Topics for Managing Generalized Bookmarks. In *WWW5 Workshop on Artificial Intelligence-based tools to help W3 users at the 5th Int'l WWW Conference, Paris, France*, 1996.

[Schmidt & Winterhalter, 2004]
Schmidt, A., Winterhalter, C. (2004): User Context Aware Delivery of E-Learning Material: Approach and Architecture. In: *Journal of Universal Computer Science (JUCS), Special Issue on Human Issues and Personalization in E-Learning,*

[Schmidt, 2004]
Schmidt, A. (2004): Kontext-Middleware zur Verwaltung dynamischer und unvollkommener Kontextinformation. In: *GI-Workshop "Grundlagen und Anwendungen mobiler Informationtechnologie". In German.*

[Schneider, 1996]
Schneider, U. (1996): Management in der wissensbasierten Unternehmung. Das Wissensnetz in und zwischen Unternehmen knüpfen. In: Schneider, U. (ed.): *Wissensmanagement: die Aktivierung des intellektuellen Kapitals.* Frankfurt am Main: FAZ. Pp. 13-48.

[Schreiber et al., 1999]
Schreiber, G., H. Akkermans, A. Anjeiwerden, R. de Hoog, N. Shadbolt, W. van de Velde, B. Wielinga (1999): *Knowledge Engineering and Management: The CommonKADS Methodology*. Cambridge / MA: MIT Press.

[Schwabe et al., 2001]
G. Schwabe, N. A. Streitz, R. Unland, eds. (2001): *CSCW-Kompendium: Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten.* Berlin, Heidelberg, New York: Springer-Verlag. *In German.*

[Schwartz, 1999]
Schwartz, David G. (1999): When Email Meets Organizational Memories: Addressing Threats to Communication in a Learning Organization. *International Journal of Human-Computer Studies* **51**(3):599-614.

[Schwarz, 2003]
Schwarz, S. (2003): Task-Konzepte: Struktur und Semantik für Workflows. In: U. Reimer, A. Abecker, S. Staab & G. Stumme (eds.): *WM 2003: Professionelles Wissesmanagement - Erfahrungen und Visionen, Beiträge der 2. Konferenz Professionelles Wissensmanagement.* LNI 28, pp. 351-356. *In German.*

[Sebastiani, 2002]
F. Sebastiani (2002): Machine Learning in Automated Text Categorization. *ACM Computing Surveys* **34**(1):1-47.

[Setten, 2001]
Setten, M. van (2001): *Personalised Information Systems.* GigaCE Project Deliverable D2.3. Twente: Telematica Instituut.

[Sindt, 2003]
T. Sindt (2003): Formal Operations for Ontology Evolution. In: *Proc. International Conference on Emerging Technologies (ICET'03).*

[Sommerlatte, 1999]
Sommerlatte, T. (1999): Marktrelevantes Wissen im Zeitalter der Informationsflut. In: C. H. Antoni and T. Sommerlatte (eds.), *Spezialreport Wissensmamagement: Wie deutsche Firmen ihr Wissen profitabel machen*, pp. 10-14, Düsseldorf: Symposium Publishing. *In German.*

[Sørensen & Snis, 2000]
Sørensen, C. & Snis, U. (2000): Codify or Collaborate? From Expert Systems to Systems of Experts for Knowledge Creation in Manufacturing. In: *Proceedings of IRIS23.* University of Trollhättan/Uddevalla, Sweden.

[Sørli et al., 1999]
Sørli, A., Coll, G.J., Dehli, E., Tangen, K. (1999): Knowledge Sharing in Distributed Organizations. In: *IJCAI-99 Workshop on Knowledge Management and Organizational Memories.*

[Speel et al., 1999]
Piet-Hein Speel, Nigel Shadbolt, Wouter de Vries, Piet Hein van Dam & Kieron O'Hara (1999): Knowledge Mapping for Industrial Purposes.

URL: http://sern.ucalgary.ca/KSI/KAW/KAW99/papers.html
[Last access: 04/18/2004]

[Spyns et al., 2002a]
P. Spyns, D. Oberle. R. Volz, J. Zheng, M. Jarrar, Y. Sure, R. Studer, R. Meersman (2002): OntoWeb – a Semantic Web Community Portal. In: [Karagiannis & Reimer, 2002].

[Spyns et al., 2002b]
Spyns, P., Meersman, R. & Jarrar, M. (2002): Data Modelling versus Ontology Engineering. *SIGMOD Record – Web Edition* **31**(4). Special Issue on Semantic

Web and Data Management.

[Staab, 2002]
Staab, S. (2002): Wissensmanagement mit Ontologien und Metadaten. *Informatik Spektrum* **25**(3):194-209. *In German.* Berlin, Heidelberg: Springer-Verlag.

[Staab & Schnurr, 2000]
Staab, S. & Schnurr, H.-P. (2000): Smart Task Support Through Proactive Access to Organizational Memory. Knowledge-Based Systems **13**(5): 251-260. Elsevier.

[Staab & Schnurr, 2002]
Staab, S. & Schnurr, H.-P. (2002): Knowledge and Business Processes: Approaching an Integration. In: [Dieng-Kuntz & Matta, 2002], pp. 75-88.

[Staab & Studer, 2003]
Staab, S. & Studer, R. eds. (2003): *Handbook on Ontologies.* Springer Series on Handbooks in Information Systems. Berlin, Heidelberg, New York: Springer-Verlag.

[Stahlknecht & Hasenkamp, 2002]
Stahlknecht, P & Hasenkamp, U. (2002): *Einführung in die Wirtschaftsinformatik*. Berlin, Heidelberg: Springer-Verlag. *In German.*

[Stein, 1995]
Stein, E. (1995): Organizational Memory: Review of Concepts and Recommendations for Management. *Int. Journal of Information Management* **15**(1):17-32.

[Stein & Zwass, 1995]
Stein, E. & Zwass, V. (1995): Actualizing Organizational Memory with Information Systems. *Information Systems Research* **6**(2):85-117.

[Steier et al., 1995]
Steier, D., Huffman, S.B. & Hamscher, W.C. (1995): *Meta-Information for Knowledge Navigation and Retrieval: What's in There.* Price Waterhouse Working Paper.

[Stojanovic & Stojanovic, 2002]
Stojanovic, N. & Stojanovic, L. (2002): Usage-oriented Evolution of Ontology-based Knowledge Management Systems. *Proc. 1st Int. Conf. on Ontologies, Databases and Application of Semantics (ODBASE-2002).*

[Stojanovic et al., 2002]
Stojanovic, L., Stojanovic, N. & Handschuh, S. (2002): Evolution of the Metadata in the Ontology-based Knowledge Management Systems. In: *German Workshop on Experience Management 2002.*

[Stojanovic, 2003a]
Stojanovic, N. (2003): On the Query Refinement in the Ontology-Based Searching for Information. In: *CAiSE 2003*, pp. 324-339.

[Stojanovic, 2003b]
Stojanovic, N. (2003): On the Role of Query Refinement in Searching for Information: The Librarian Agent Query Refinement Process. In: *4th Int. Conf. on Web Information Systems Engineering (WISE 2003),* pp. 41-52. IEEE Computer Society.

[Strohmaier, 2003]
Strohmaier, M. (2003): A Business Process Oriented Approach for the Identification and Support of Organizational Knowledge Processes. In: *Proc. 4. Oldenburger Fachtagung Wissensmanagement, Potenzial – Konzepte – Werkzeuge.*

[Studer et al., 1999]
Studer, R., Abecker, A. & Decker, S. (1999): Informatik-Methoden für das Wissensmanagement. In: *Festschrift zum 60. Geburtstag von Prof. Dr. Wolffried Stucky.* Stuttgart: Teubner Verlag. *In German.*

[Stumme et al., 2003]
Stumme, G. et al. (2003) : *The Karlsruhe View on Ontologies.* Technical Report, University of Karlsruhe, Institute AIFB.

[Sumner et al., 1999]
Sumner, T., Domingue, J., Zdrahal, Z., Millican, A. & Murray, J. (1999): Moving from On-the-Job Training towards Organisational Learning. *Proc. 12th Banff Knowledge Acquisition Workshop.*

[Sure, 2003]
Sure, Y. (2003): *Methodology, Tools and Case Studies for Ontology based Knowledge Management.* Dissertation, University of Karlsruhe, Department of Economics and Business Engineering.

[Tautz, 2002]
Tautz, Ch. (2002): The Role of Ontologies and Taxonomies in Knowledge Technologies. Talk at *Knowledge Technologies*, Seattle / USA, March 2002.

[Thiesse, 2001]
Thiesse, F. (2001): *Prozessorientiertes Wissensmangement: Konzepte, Methode, Fallbeispiele.* Dissertation Universität St. Gallen, HSG. Bamberg: Difo-Druck GmbH. *In German.*

[Tönshoff et al., 2001]
Tönshoff, H.K., Apitz, R., Lattner, A.A. & Schlieder, C. (2001): KnowWork – An Approach to Coordinate Knowledge within Technical Sales, Design and Process Planning Departments. In: Thoben, K.D., Weber, F. & Pawar, K.S. (eds.), *7th Int. Conference on Concurrent Enterprising (ICE-01).*

[Tönshoff et al., 2002]
Tönshoff, H. K; Apitz, R.; Lattner, A.D. & Schäffer, C. (2002): Support for Different Views on Information in Concurrent Enterprises. In: *Proc. 8th Int. Conference on Concurrent Enterprising,* pp. 143-150.

[Trittmann, 2001]
Trittmann, R. (2001): The Organic and the Mechanistic Form of Managing Knowledge in Software Development. In: K.-D. Althoff, R.L. Feldmann, W. Müller (eds.): *Advances in Learning Software Organizations.* Berlin, Heidelberg, New York: Springer-Verlag. Pp. 22-36.

[Tschaitschian et al., 1997]
Tschaitschian, B., Abecker, A. & Schmalhofer, F. (1997): Information Tuning with KARAT: Capitalizing on Existing Documents. In: E. Plaza & R. Benjamins (eds.) *10th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW-97).* Berlin, Heidelberg, New York: Springer-Verlag, LNAI 1319.

[Uschold & King, 1995]
Uschold, M. & King, M. (1995): Towards a Methodology for Building Ontologies. In: *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing.*

[Uschold et al., 1998]
Uschold, M., King, M., Moralee, S. & Zorgios, Y. (1998): The Enterprise Ontology. *The Knowledge Engineering Review,* Vol. 13.

[VDI, 1992]
VDI (1992): *VDI-Richtline 5006: Bürokommunikation; Expertensysteme in betriebswirtschaftlichen Anwendungen. In German.*

[Vicari et al., 1996]
Vicari, S., Krogh, G. von, Roos, J. & Mahnke, V. (1996): Knowledge Creation through Cooperative Experimentation. In: Krogh, G. von & Roos, J., eds.: *Managing Knowledge. Perspectives on Cooperation and Competition.* Thousand Oaks: Sage. Pp. 184-202.

[Wainer & Braga, 2001]
Wainer, J. & Braga, D.P. (2001): Symgroup: Applying Social Agents in a Group Interaction System. In: Proc. *2001 Int. ACM SIGGROUP Conf. on Supporting Group Work.*

[Walsh & Ungson, 1991]
Walsh, J.P. & Ungson, G.R. (1991): Organisational Memory. *Academy of Management Review* **16**(1):57-91.

[Wargitsch, 1997]
Wargitsch, C. (1997): WorkBrain: Merging Organizational Memory and Workflow Management Systems. In: [Abecker et al., 1997].

[Wargitsch et al., 1998]
Wargitsch, C., Wewers, T. & Theisinger, F. (1998): An Organizational-Memory-Based Approach for an Evolutionary Workflow Management System – Concepts and Implementation. In *Proceedings HICCS`31, Vol. 1,* pp. 174-183.

[Watson, 1996]
Watson, R.T. (1996): *Data Management. An Organizational Perspective.* New York et al.

[Watson, 2002]
Watson, I., ed. (2002): *Applying Knowledge Management: Techniques for Building Corporate Memories.* San Francisco: Morgan Kaufmann Publishers.

[Weber et al., 2001]
Weber, R., Aha, D.W. & Becerra-Fernandez, I. (2001): Intelligent Lessons Learned Systems. *International Journal of Expert Systems Research & Applications* **20**(1):17-34.

[Weih, 2002]
Weih, T. (2002): *Gestaltungsrahmen und Vorgehensmodell zur Einführung von CRM auf der Grundlage eines prozessorientierten, wissensbasierten Ansatzes.* Diplomarbeit. Stuttgart: FhG IAO*. In German.*

[Welty, 1996]
Welty, C. (1996): Intelligent Assistance for Navigating the Web. In *Proc. Of the 1996 Florida AI Research Symposium*.

[Wenzel & Maus, 2001]
Wenzel, C. & Maus, H. (2001): Leveraging Corporate Context Within Knowledge-Based Document Analysis and Understanding. *Int. Journal on Document Analysis and Recognition* **3**(4):248-260. Special Issue on Document Analysis for Office Systems. Berlin, Heidelberg, New York: Springer-Verlag.

[Wenger, 1998]
Wenger, E. (1998): *Communities of Practice. Learning, Meaning and Identity.* Cambridge: Cambridge University Press.

[Wess, 1996]
Wess, S. (1996): *Fallbasiertes Problemlösen in wissensbasierten Systemen zur Entscheidungsunterstützung und Diagnostik.* Volume 126 Dissertationen zur Künstlichen Intelligenz. Berlin: Akademische Verlagsgesellschaft Aka GmbH. *In German.*

[WfMC, 1995]
Workflow Management Coalition (1995): *The Workflow Reference Model.* Document number TC00-1003. Issue 1.1.

[WfMC, 1999]
Workflow Management Coalition (1999): *Workflow Management Coalition Terminology & Glossary.* Document number WFMC-TC-10011. Issue 3.0.

[WfMC, 2001]
Workflow Management Coalition (2001): *Workflow Process Definition Interface – XML Process Definition Language.* Document number WFMC-TC-10025.

[Wiederhold, 1992]
Wiederhold, G. (1992): Mediators in the Architecture of Future Information Systems. *IEEE Computer* **25**(3):38-49.

[Wiederhold & Genesereth, 1997]
Wiederhold, G. & Genesereth, M.R. (1997): The Conceptual Basis for Mediation Services. *IEEE Expert* **12**(5): 38-47.

[Wijnhoven, 1998]
Wijnhoven, F. (1998): Designing Organizational Memories: Concept and Method. *Journal of Organizational Computing and Electronic Commerce* **8**(1):29-55.

[Willke, 1996]
Willke, H. (1996): Dimensionen des Wissensmanagements - Zum Zusammenhang von gesellschaftlicher und organisationaler Wissensbasierung. In: G. Schreyögg & P. Conrad (eds.): *Managementforschung 6*. Berlin. Pp. 263-304. *In German.*

[Winograd & Flores, 1987]
Winograd, T. & Flores, F. (1987): *Understanding Computers and Cognition.* Addison-Wesley.

[Woitsch & Karagiannis, 2003]
Woitsch, R. & Karagiannis, D. (2003): Process-Oriented Knowledge Management Systems Based on KM-Services: the PROMOTE® Approach*. Int. Journal of Intelligent Systems in Accounting, Finance & Management* **11**(4):253-267. John Wiley & Sons, Inc.

[Wolf et al., 1999]
Wolf, T., Decker, S. & Abecker, A. (1999): Unterstützung des Wissensmanagements durch Informations- und Kommunikationstechnologie. In: *WI'99 – 4. Internationale Tagung Wirtschaftsinformatik.* Heidelberg: Physica-Verlag. *In German.*

[Wooldridge & Ciancarini, 2001]
Wooldridge, M. & Ciancarini, P. (2001): Agent-Oriented Software Engineering: The State of the Art. In: P. Ciancarini & M. Wooldridge (eds.): *Agent-Oriented Software Engineering - AOSE 2000.* Berlin, Heidelberg, New York: Springer-Verlag. LNCS 1957, pp. 1-28.

[Wooldridge & Jennings, 1995]
Wooldridge, M. & Jennings, N.R. (1995): Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review* **10**(2):115-152.

[Yimam, 2000]
Yimam, D. (2000): Expert Finding Systems for Organizations: Domain Analysis and the DEMOIR Approach. In: Ackerman M., Cohen A., Pipek V. & Wulf V. (eds): *Beyond Knowledge Management: Sharing Expertise.* Cambridge: MIT Press.

[Yimam-Seid & Kobsa, 2003]

Yimam-Seid, D. & Kobsa, A. (2003): Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. *Journal of Organizational Computing and Electronic Commerce* **13**(1):1-24.

[Young, 1998]
Young, R. (1998): *KM Fundamentals.* Cambridge: Knowledge Associates Ltd.

[Zambonelli et al., 2003]
Zambonelli, F., Jennings, N.R. & Wooldridge, M. (2003): Developing Multiagent Systems: The Gaia Methodology. *ACM Trans. Softw. Eng. Methodol.* **12**(3): 317-370.

[Zdrahal et al., 2000]
Zdrahal, Z., Mulholland, P., Domingue, J. & Hatala, M. (2000): Sharing Engineering Design Knowledge in a Distributed Environment. *Journal of Behaviour and Information Technology* **19**(3):189-200.

[Zigurs & Buckland, 1998]
Zigurs, I & Buckland, B.K. (1998): A Theory of Task/Technology Fit and Group Support Systems Effectiveness. *Management Information Systems Quarterly* **22**(3):313-334.

General acknowledgement:

These thesis couldn't have been accomplished without financial support from many public and industrial research funding institutions. To mention the most important ones:

- The German Federal Ministry for Research and Education bmb+f funded the basic research project KnowMore (Knowledge Management for Learning Organizations) under grant ITW 9705/3.
- The European Commission co-funded the RTD project Know-Net (Knowledge Management with Intranet Technologies) under grant EP28928.
- The German Federal Ministry for Research and Education bmb+f funded the basic research project FRODO (A Framework for Distributed Organizational Memories) under grant 01 IW 901.
- The European Commission co-funded the European RTD project DECOR (Delivery of Context-Sensitive Organizational Knowledge) under grant IST-1999-13002 and INKASS (Intelligent Knowledge Asset Sharing and Trading) under grant IST-2001-33373.
- The Forschungszentrum Informatik FZI an der Universität Karlsruhe where I had the opportunity to compile this thesis, partially funded by the European projects Ontologging and SWWS.

„So eine Arbeit wird eigentlich nie fertig,
man muß sie für fertig erklären
wenn man nach Zeit und Umständen
das Mögliche getan hat.“

Johann Wolfgang von Goethe