Context-Aware, Proactive Delivery of Task-Specific Information: The KnowMore Project

Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann,
Otto Kühn, and Michael Sintek

German Research Center for Artificial Intelligence (DFKI) GmbH

P.O.Box 2080, D-67608 Kaiserslautern, Germany
Phone:+49 631 205 3582, Fax:+49 631 205 3210

{aabecker,bernardi,hinkelma,kuehn,sintek}@dfki.uni-kl.de

Abstract

From an IT point of view, a key objective of successful knowledge management is to provide relevant and necessary information at the right time to support humans in accomplishing their tasks. This paper presents a prototypical system which meets this objective in an enterprise environment. Based on context information associated with the enterprise's business processes, an integration of workflow engine and information assistant enables active presentation of relevant information to the user. We describe the functionality of the system and elaborate (i) on necessary extensions to the business process models, (ii) the ontologies used for information modeling, and (iii) the integration of workflow engine and active information assistant. The prototype system has been developed in the KnowMore project of the DFKI Knowledge Management Group.

Keywords: organizational memory, business-process oriented knowledge management, cooperative information system

Running title: Context-Aware, Proactive Delivery of Task-Specific Information

1 Introduction

Corporate knowledge management (KM) has been recognized by many companies as a very important objective. Managing a company's knowledge assets comprises, among other things, (i) managing human resources, i.e. the organization's employees, in such a way that they construct, renew, and try to communicate and share their *tacit knowledge*, and (ii) managing the company's *explicit knowledge*, i.e. documents, formal rules, etc.—technically, *information*—in such a way that the right portion of explicit information available in the "organizational memory" is provided to the right person at the appropriate time.

Having a look at existing IT support for Knowledge Management, one can identify two main approaches [Kühn and Abecker, 1998] which roughly correspond to the above two KM dimensions:

The process-centered view mainly understands KM as a social communication process which can be improved by various aspects of *groupware support*. It is based on the observation that the most important knowledge source in an enterprise are the employees and that solving *wicked problems* [Conklin and Weil, 1997] is merely a process of achieving social commitment than one of problem solving. Basic techniques for this approach try to stimulate human communication and collaboration, and come from Computer-Supported Cooperative Work (CSCW) and from Workflow Management (cp. [Prinz and Syri, 1998], [Simone and Divitini, 1998]).

The product-centered view focuses on knowledge documents, their creation, storage, and reuse in computer-based *organizational memories*. It is based on the idea of explicating, documenting, and formalizing knowledge to have it as a tangible ressource, and trying to present the right information sources at the appropriate time to the user. Basic techniques come from Document Management, Knowledge-Based Systems, and Information Systems (cp. [Kühn and Abecker, 1998], [van Heijst *et al.*, 1998]).

In this paper we mainly focus on the latter approach, namely intelligent information management. However, in order to justify the terms "knowledge management" and "organizational memory" instead of "information management" and "information system", we focus on aspects motivated by the analogy to the human memory, as well as by practical hurdles for the introduction of KM systems described in [Kühn and Abecker, 1998]. Two important aspects we address with our approach, are the *proactive* delivery of information to the user by an intelligent information assistant, and the idea of assessing *task context* information to improve information

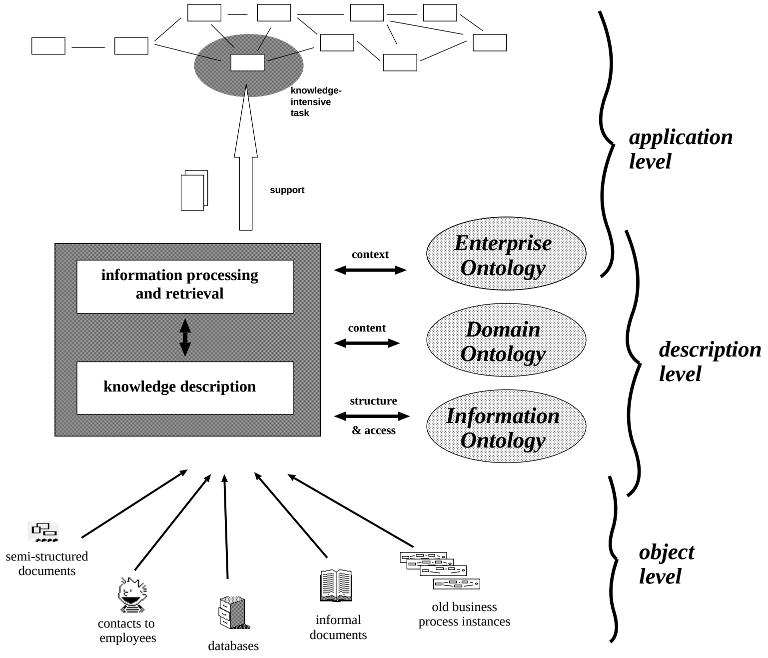


Figure 1: *KnowMore supports complex tasks by context-specific information supply.*

retrieval.

The system which we describe realizes a novel, purposeful integration of existing, fielded techniques, namely Workflow Management and Knowledge-Based Information Retrieval. The basis of our approach is the observation that business process models in an enterprise can provide worthful context information for highly-selective, situation-specific information supply. So, we integrate workflow engine and information assistant such that useful and necessary information is actively presented to support the human user in performing a knowledge-intensive activity. This approach has been prototypically implemented by the DFKI Knowledge Management Group in the KnowMore project (see also [Abecker *et al.*, 1999a, Abecker *et al.*, 1998b]).

The overall approach of our KnowMore system is sketched in Figure 1: while a human user is always in the loop of a running business process, accomplishing knowledge-intensive tasks (KITs) is supported by information retrieved from the complete organizational knowledge base. All kinds of information are treated in a similar way thus giving access to databases and retrieving informal documents, as well as offering communication paths to knowledgeable colleagues. Homogeneous processing and retrieval is based upon knowledge-based ontological information modeling. The running business process is exploited not only for automatically launching queries to the OM[1], but also to define situation-specific context for enabling a concise retrieval of relevant knowledge items.

Before discussing the technical framework for realizing such a functionality, Section 2 will give an impression of the system prototype with the help of an example. The necessary extensions to the business process models, the ontologies used for information modeling, and the integration of workflow engine and information assistant are described in the subsequent Sections 3, 4, and 5. After some remarks on implementation issues (Section 6), we discuss some related and its impact on possible future work in Section 7, and finally conclude with Section 8.

2 A Quick Run Through the KnowMore System Functionality

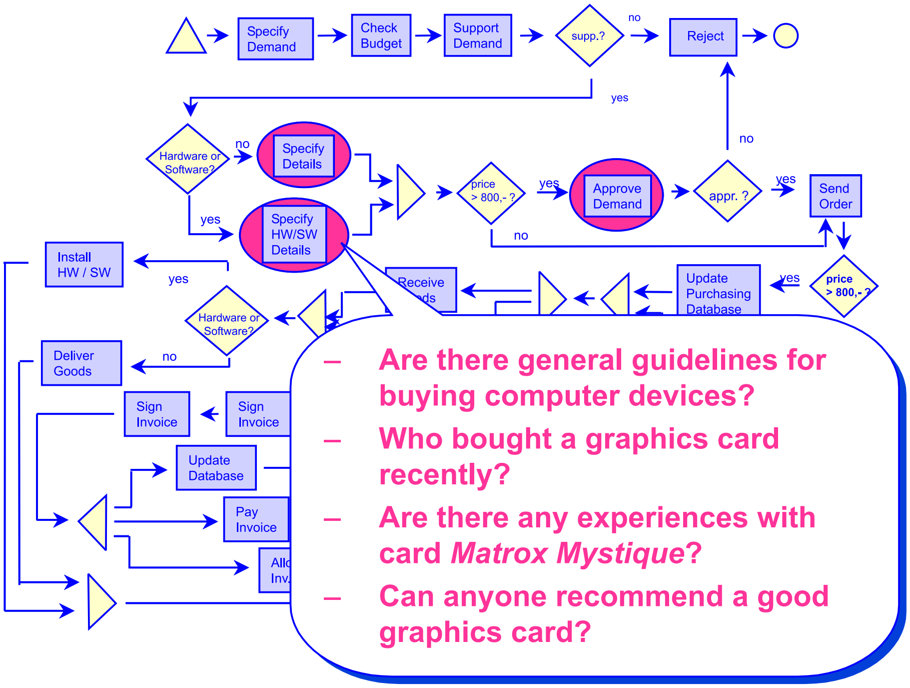
As a running example consider the business process of purchasing goods for our research institute DFKI (similar business processes can be found in most public administrations). Figure 2 gives an impression of the formal workflow to be performed to this end. It is graphically represented with the ADONIS business process modeling tool (cp. [Karagiannis *et al.*, 1996, BOC GmbH, 1998]).

A purchasing process starts with an employee filling out a demand specification form and mainly consists of a fairly deterministic sequence of more or less simple administrative steps like checking the budget, writing the order, or assigning an inventory number.

However, among such simple administrative activities, there are a few working steps requiring expert knowledge and purchasing experience. Some of them are marked in the picture by a dark surrounding circle. We will focus here on the preparation of a detailed 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!"*). If an unexperienced employee should accomplish such a detailed demand specification, questions like the ones shown in Figure 2 could arise, the answering of which would be a

---

[1] Since we address only technical support for Knowledge Management, we will use synonymously the terms OM and OMIS, for Organizational Memory and Organizational Memory Information System, throughout this paper. But please keep in mind that implementing KM and OM solutions in an organizational application context must cover manifold non-technical issues, such as, e.g., organizational aspects of KM processes.



- Are there general guidelines for buying computer devices?
- Who bought a graphics card recently?
- Are there any experiences with card *Matrox Mystique*?
- Can anyone recommend a good graphics card?

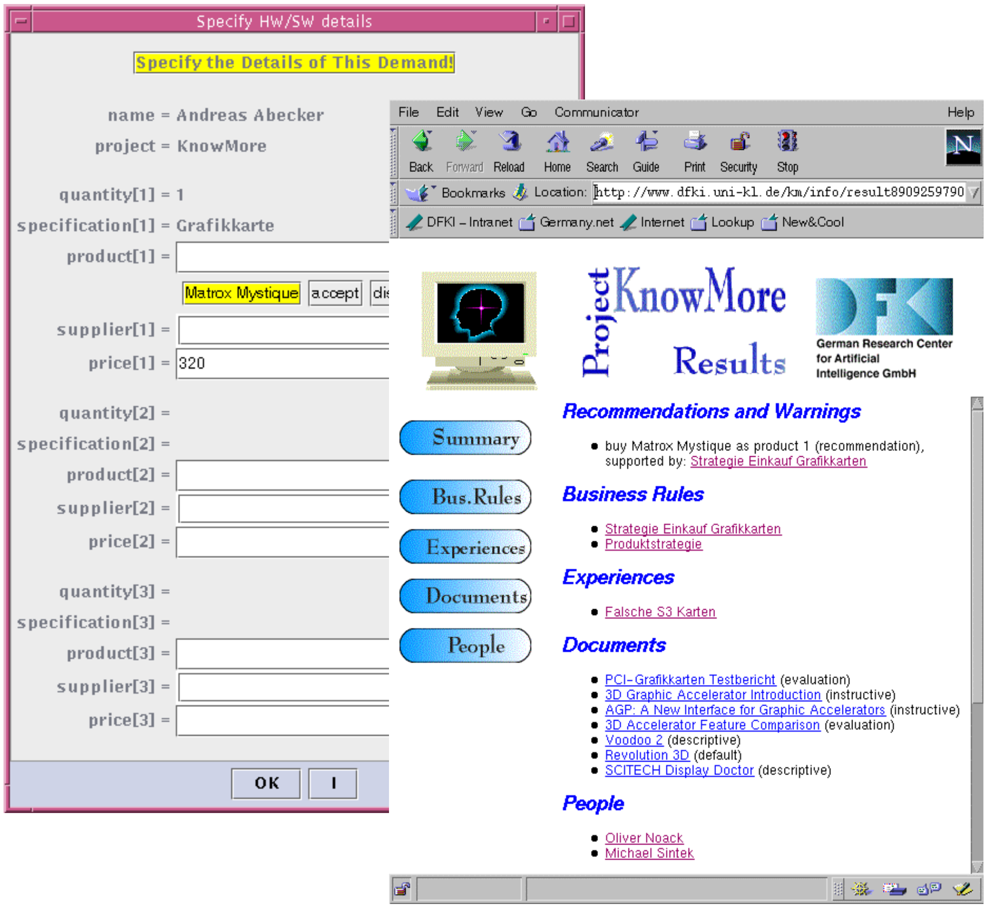
Figure 2: *Knowledge-intensive tasks are often embedded into "tame" workflows.*

helpful service of an Organizational Memory Information System.

The KnowMore poject provided a framework and toolbox which helps to easily build applications which would actively offer to the user answers to such questions as described above. The KnowMore approach proceeds with the following steps:

1. During *business process modeling*:
   - The application programmer models the overall business process with a conventional BPM / workflow tool.
   - For knowledge-intensive tasks (KITs), the respective workflow activities are extended by generic queries to be posed to the OM, the answers of which would help to perform these tasks.
2. During *workflow enactment*, the system instantiates the generic 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.

Figure 3 shows a screenshot of our system prototype. On the left, in the background, we see an editor window of the workflow application used to create detailed



Specify the Details of This Demand!

name = Andreas Abecker
project = KnowMore
quantity[1] = 1
specification[1] = Grafikkarte
product[1] = Matrox Mystique
supplier[1] =
price[1] = 320
quantity[2] =
specification[2] =
product[2] =
supplier[2] =
price[2] =
quantity[3] =
specification[3] =
product[3] =
supplier[3] =
price[3] =

Location: http://www.dfki.uni-kl.de/km/info/result8909259790

Recommendations and Warnings

- buy Matrox Mystique as product 1 (recommendation), supported by: Strategie Einkauf Grafikkarten

Business Rules

- Strategie Einkauf Grafikkarten
- Produktstrategie

Experiences

- Falsche S3 Karten

Documents

- PCI-Grafikkarten Testbericht (evaluation)
- 3D Graphic Accelerator Introduction (instructive)
- AGP: A New Interface for Graphic Accelerators (instructive)
- 3D Accelerator Feature Comparison (evaluation)
- Voodoo 2 (descriptive)
- Revolution 3D (default)
- SCITECH Display Doctor (descriptive)

People

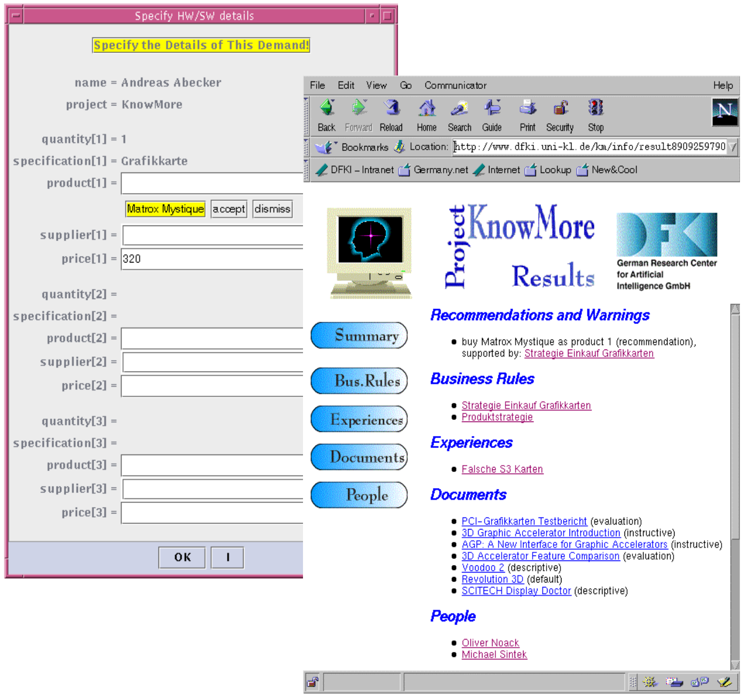
- Oliver Noack
- Michael Sintek

Figure 3: *The KnowMore system offers context-sensitive information supply.*

demand specifications. The input mask accepts up to three items to be purchased and already contains the initial specification given by the end user (he needs one graphics card, in German *Grafikkarte*). Now, it has to be decided which concrete card to buy (the `product` slot in the input mask), and from which supplier (the `supplier` slot in the input mask).

The KnowMore system supports this decision in the following way:

At the moment when the workflow engine starts this activity, the system takes the information needs associated to the activity at modeling time, and finds out whether some element of the OM can already compute a decision suggestion (i.e., whether there is some expert system or decision support functionality available which can readily be evaluated). This suggested decision value is inserted in the user input mask offering a proposed solution (in the example, the suggestion is to buy a `Matrox Mystique` card).



Specify the Details of This Demand!

name = Andreas Abecker
project = KnowMore

quantity[1] = 1
specification[1] = Grafikkarte
product[1] =
Matrox Mystique
supplier[1] =
price[1] = 320

quantity[2] =
specification[2] =
product[2] =
supplier[2] =
price[2] =

quantity[3] =
specification[3] =
product[3] =
supplier[3] =
price[3] =

http://www.dfki.uni-kl.de/km/info/result8909259790

Recommendations and Warnings

- buy Matrox Mystique as product 1 (recommendation), supported by: Strategie Einkauf Grafikkarten

Business Rules

- Strategie Einkauf Grafikkarten
- Produktstrategie

Experiences

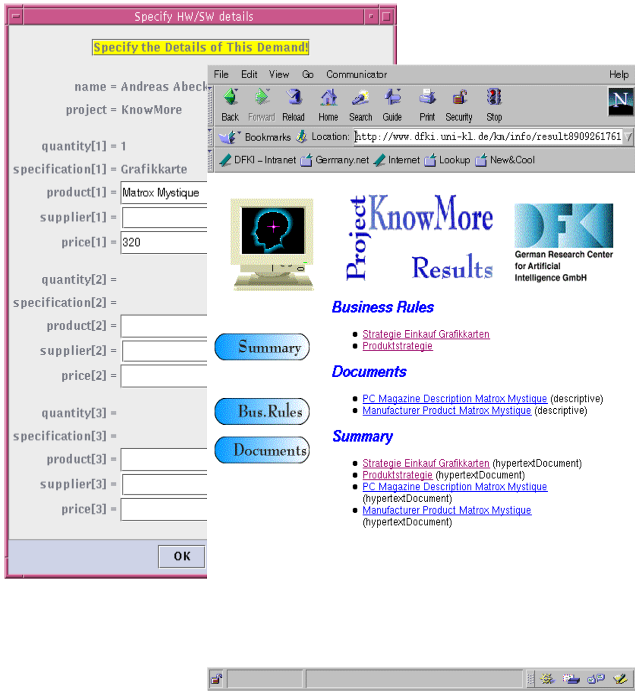
- Falsche S3 Karten

Documents

- PCI-Grafikkarten Testbericht (evaluation)
- 3D Graphic Accelerator Introduction (instructive)
- AGP: A New Interface for Graphic Accelerators (instructive)
- 3D Accelerator Feature Comparison (evaluation)
- Voodoo 2 (descriptive)
- Revolution 3D (default)
- SCITECH Display Doctor (descriptive)

People

- Oliver Noack
- Michael Sintek



Specify the Details of This Demand!

name = Andreas Abeck
project = KnowMore

quantity[1] = 1
specification[1] = Grafikkarte
product[1] = Matrox Mystique
supplier[1] =
price[1] = 320

quantity[2] =
specification[2] =
product[2] =
supplier[2] =
price[2] =

quantity[3] =
specification[3] =
product[3] =
supplier[3] =
price[3] =

http://www.dfki.uni-kl.de/km/info/result8909261761

Business Rules

- Strategie Einkauf Grafikkarten
- Produktstrategie

Documents

- PC Magazine Description Matrox Mystique (descriptive)
- Manufacturer Product Matrox Mystique (descriptive)

Summary

- Strategie Einkauf Grafikkarten (hypertextDocument)
- Produktstrategie (hypertextDocument)
- PC Magazine Description Matrox Mystique (hypertextDocument)
- Manufacturer Product Matrox Mystique (hypertextDocument)

Figure 4: *Changes in the process status result in refined information support.*

Further, the system determines about which decision variables it is able to offer some information, and it inserts information buttons (“`I`”) at the appropriate places in the input mask. If the user wants some supporting information for one of these decisions, pressing the “`I`”-Button starts a query to the OM system. Such a query may retrieve several classes of information. It might yield highly recommended company-internal business rules for purchasing in general or for specific product classes, or technical information about possible buying alternatives; it might also produce pointers to knowledgeable colleagues who are known to be competent for such kinds of decision—because of their entry in the personal skill database, because of their training records or position in the company, or because they recently did a similar purchase. The manifold different answers will have different relevance in the concrete situation. So they are ordered according to the relevance estimation computed by the retrieval function, taking into account a predefined order (specified in the postprocessing rules of the support specification) based upon the task-specific relevance of different information types. The ordered list of results is offered to the user as a list of hyperlinks in the KnowMore information browser (Figure 3, right side, foreground).

The user either accepts or overwrites a suggested solution. When the partial demand specification is more detailed, or when the situation changes for any reason, he or she might ask again for KnowMore support via an “`I`” button. The system then notices the change of the state of affairs (technically, this means, that some variable value has been modified) and reevaluates the query against the archive system. Figure 4 shows the effect of taking into account the change in the process state: in the right part of the picture, the user has selected the `Matrox Mystique` card which considerably narrows down the search in the OM. Now, all documents are eliminated which have no direct relationship to this specific card. What remains are only compulsory purchasing business rules and 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`, whereas in the previous process state (left hand side of Figure 4) all suppliers would be described which sell graphics cards in general.

Research topics for realizing the KnowMore functionality

For realizing the described functionality, the KnowMore system makes the following technical provisions:

- Each KIT in a business process model is equipped with a *support specification* describing the respective *information needs* as generic queries or query schemes to be instantiated at runtime, together with their appropriate preconditions and postprocessing rules for the results. For this, we have to describe the appropriate representation means.

- In order to instantiate the query schemes at runtime thus exploiting situation-specific knowledge and context parameters, the retrieval process must have access to the workflow parameters, and business process models must be enriched by so-called *KIT variables.* These variables describe the information flow between tasks in the workflow, and are the communication channel between workflow and information retrieval agents (in the above example, the `product` slot). They are embedded into a domain ontology (essentially, this means that their values must be of a type defined as an ontology concept).

  The modeling of support specifications, information needs, and KIT variables is described in Section 3.

- For enabling precise-content retrieval from manifold heterogeneous sources in the OM, a powerful representation scheme for *uniform knowledge description* must be provided. In KnowMore, we model structure and metadata, information content, and information context of OM knowledge items on the basis of formal ontologies.

  This is discussed in Section 4.

- Within an information agent, the realization of *ontology and metadata-based retrieval* with background knowledge and search heuristics constitutes a complex interaction of formal inferences and informal retrieval.

  Section 5 gives an impression of the implemented information retrieval approach as well as some hints for future extensions.

3 Modeling Information Needs

Knowledge-intensive tasks (KITs) are at the heart of any knowledge work which in turn can be increasingly identified inside the core competencies of modern enterprises [Shum, 1998, Conklin and Weil, 1997, Davenport *et al.*, 1996]. Working on KITs is not problem-solving in the proper sense (which means performing a well-defined sequence of steps leading to a clear output); it is merely characterized by unforeseeable communication, negotiation, and control flow, and ususally stops with sort of "social agreement" achieved within the project team. A central property of all activities involved is the extensive dealing with acquisition, creation, packaging, and application of knowledge in manifold forms. Normally, such tasks are represented as manual activities in workflow management systems which are in principle beyond the scope of automated support.

In contrast, the basic assumption of the KnowMore approach is that KITs can be described (at least partly) by (one or more) explicitly expressible *information needs* which need to be satisfied for achieving the goals of the task. Fulfilling an information need may require actions of varying complexity, ranging from database queries using well-defined selections, over access to relevant documents from the corporate library, extended keyword retrieval on the world-wide web, up to queries passed to an expert system.

A KIT forms a unit of sense, but further details might be given by specifying several (partial) information needs. Each information need will result in some information which supports a particular aspect of the complete KIT. If there are logical or time dependencies among the output of several information needs known at process definition time, they can be represented in the KIT description. To this end, the KnowMore system provides *preconditions* in the various information needs, and *processing rules* for their results. Both influence the way the information needs are interpreted and fulfilled during process execution. If all relationships and interdependencies between information needs were known, we could probably represent the KIT as an ordinary workflow. But this is *per definitionem* not possible for a KIT. Hence, the imposed structure is only partial; it will be used for more effective information search and presentation, and not for guiding and controlling KIT processing.

Technically, a KIT is a special case of an ordinary workflow activity extended by a *support specification* (containing information needs and processing rules) which may refer to the global and local process context (the lower part of the *task* box in Figure 5). The support specification inside a KIT representation employs a description frame as shown in Figure 6. This description frame specifies:

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, or: if a specific parameter is already known, skip this need) or on the state of the process (skip if time is critical).

The **agent-spec** description of the relevant information is interpreted as a re-

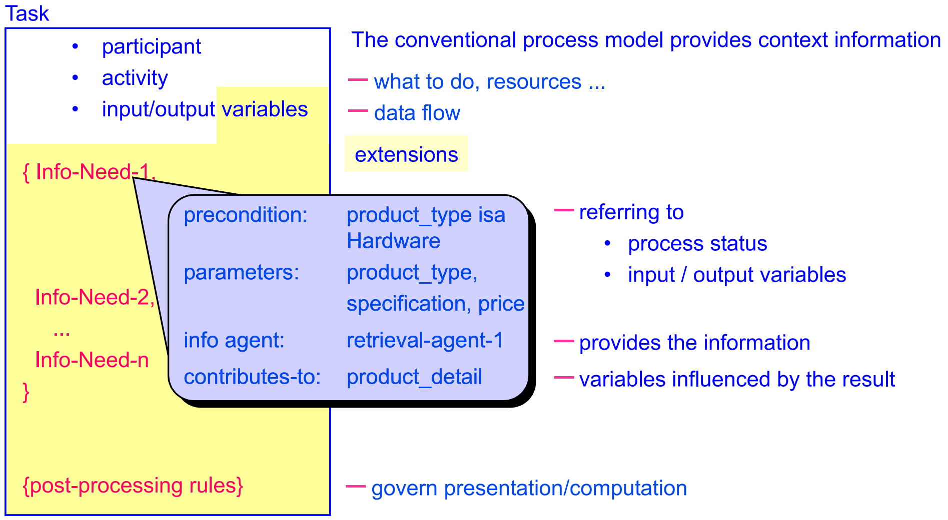


Figure 5: *Information needs refer to the current global and local process status.*

mote procedure call to a specific information agent. Such a software agent is responsible for retrieving relevant information. At runtime, it is invoked and provided with a number of *parameters* taking context information from the actual working situation to the retrieval process.

The **from** description of the information sources is a construct occuring in the current KnowMore implementation which might become obsolete (or, an optional parameter) in future versions of the system. In principle, determining the info sources which are relevant for a particular information need is a central objective of the information agent. By computing $info\text{-}source = f\ (parameters, expected\text{-}output, callingActivity, processInstance)$, the agent finds the knowledge source according to the goal and context information. As a first step, however, we identify suitable info sources at process definition time, e.g., the well-known databases of the enterprise. Thus *from* contains a list of relevant info sources.

The **contributes-to** field indicates the goal of the particular info need: its purpose is to support the user in finding values for the variables mentioned here. On the basis of this information, the interconnection between different information needs can be deduced and evaluated by the system.

The **processing** rules govern the postprocessing of the retrieved information. Usually, the result of evaluating the information needs is presented to the user. In certain cases, however, it is possible to specify further operations (e.g., a formal knowledge item is used for direct computation by some algorithm). The result of some information need—seen in meta-information from the information assistant

Context information, inherited from simple task	
name,	a symbol identifying the KIT
execute,	denotes the application which is supported by the KIT
input:{variable}	denotes the local context of the KIT
output:{variable}	denotes the local goal of the KIT
Process context, provided by the runtime environment	
#callingActivity, #processInstance	These symbols may be used in the following to refer to the relevant workflow control data. These symbols indicate the actual instances of the activity which initiated the KIT and of the process which contains the activity. This information is provided at runtime by the workflow management system.
Support specification, containing a set of information needs which detail the KIT and connect between interface variables and information retrieval queries	
local-variables:{variable}	declaration of additional variables used in the KIT description
infoneeds:{	
(name,	a symbol
description,	a comment
precondition:{ *constraint-object*}	a set of constraints on any of the variables accessible from inside the KIT. The information need is only evaluated if these preconditions are fulfilled.
agent-spec,	a string, containing the specification of the information needed which can be given at process definition time.
parameters:{variable},	a subset of the relevant input variables, local variables, or the above-mentioned symbols denoting references to the calling activity and the process instance. The values of these interface elements are only known at process execution time.
from:*info-source-description*,	a set of symbols denoting info sources. This might be omitted, if the knowledge agent which processes the KIT is able to compute the relevant information sources.
contributes-to:{variable}	local variables or output variables which are filled using the result of this info need.
) }	
processing:{ if *constraintobject* do *action* }	A set of forward rules working on the result of the information needs.

Figure 6: *Support specification for a knowledge-intensive task.*

system—can also trigger further operations. The *constraintobject* may contain expressions about any variable accessible inside the KIT or about meta-information which is provided by the system. Examples of meta-information are, e.g., *empty result* or *count of produced information objects. Action* comprises sorting of results, calculation of values, the setting of variables, or the activation of derived information needs.

KIT modeling in the purchasing example. Coming back to our DFKI purchasing example (see Section 2, Figure 2), we focus on the detailed demand specification mentioned above (see Figure 7).

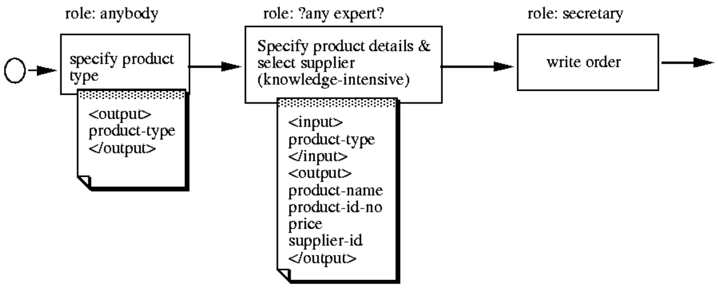


Figure 7: *A section from a purchasing process.*

The process starts when an employee indicates the need for some product, e.g., some hardware component. The details of this product are to be specified by experts in this field. As numerous questions of compatibility, availability, and bargain must be considered, this is a difficult and knowledge-intensive task. If the specification has been fixed and the supplier is known, the purchasing department secretary may write the order, and the process goes on.

In order to provide adequate support we model the knowledge-intensive task in more detail. As already pointed out in Section 3 we can assume various information sources:

- Names of products, their technical data, and additional technical information exists in some database (e.g., the PC-SHOPPING list on the WWW).
- The product ID number and the price of the product are taken from catalogues of the supplier.
- A list contains all suppliers in contact with the enterprise.

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

Figure 8: *A KIT representation for the purchasing example.*

- Various people are specialists on some–or–the–other details about particular products. Depending on the actual product to be purchased, they should be asked.
- Previous experiences about products and suppliers are collected as written notes.
- Technical details about product quality, tests, and product comparisons can be found in HTML files on the WWW, e.g., provided by computer magazines.
- Company-specific business rules are available in text form and can sometimes—like the purchasing strategy for graphics cards—be partially formalized for automatic decision support.

Figure 8 shows the respective KIT is represented as it is implemented in our current system prototype.

Cooperation between workflow system and the KnowMore assistant system. The central instance to work on the KIT is the human workflow participant. He is responsible for solving the problem at hand. Thus the worklist handler simply presents to the workflow participant an editor window with the KIT name and the input and output variables. The human user accomplishes the task at hand by filling the output variables.

In parallel, the KIT representation is passed to the KnowMore assistant system which evaluates the information needs and instantiates the parameters. It then presents the various information needs as *support offers* (e.g., "`I`" buttons) to the user, using the name and the comment of the information needs (cf. Figure 9).

The user selects interesting offers. Then, the knowledge agent determines the relevant information sources, creates suitable queries from the information need and performs the information retrieval. The result is presented as supporting information to the user (cf. Figure 3).

Any change in the various variables which the user has to fill must result in a re-evaluation of the information needs depending on these variables. Again, this shall be realized as a suggestion to the user: The previous results are marked as *possibly outdated*, but the activation of a new retrieval process is left to the user.

As soon as the user completes the task and the filling of output variables, a message is passed to the worklist handler. Automatically, the information assistant receives a *close* signal for this particular KIT, closes the display windows under its responsibility, and exits.

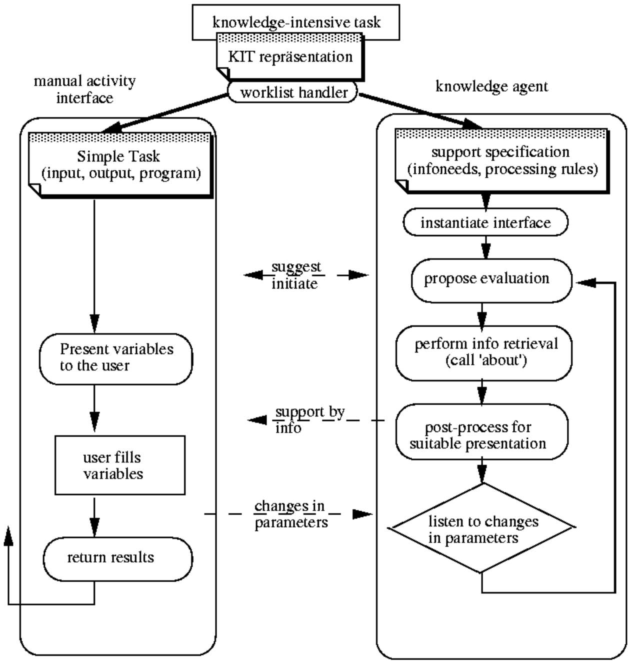


Figure 9: *KIT processing by worklist handler and KnowMore assistant system.*

In summary, the information needs modeled in the KIT representation are satisfied by the information assistant (called "knowledge agent" in Figure 9), under control of the human workflow participant, and in close interaction with the worklist handler. Thus, the integration of the knowledge agent into existing workflow environments only requires (i) the extension of the business process definition by the KIT representation, (ii) the KIT variables, and (iii) the implementation of a suitable worklist handler. The remaining workflow enactment services stay untouched.

In the support specification (see Section 3, Figures 5 and 6) there is also some postprocessing defined which determines the integration of info agent results into the overall support scenario, as well as the play-together of several info agents. In the implemented purchasing example illustrated by Figures 3 and 4, two simple kinds of postprocessing were used depending of the type of output delivered by the info agents:

1. *Computed values:* since there exists a formal decision rule in our company stating that—if there has not been an important change in technology—it should always be bought the same card as at the last buys, it can be derived a concrete purchasing suggestion, namely the `Matrox Mystique`. Such a computed value for a decision variable can directly be inserted into the user

interface of the variable editor (generally speaking, the computed value can be further processed by the *application* of the current business process activity).

2. *Informal material:* all other material (test reports, technical data sheets, memos of colleagues) has to be presented to the user in the *information browser.* Before, the postprocessing rules govern in which order to sort the results.

4 Modeling OM Sources

A uniform, intelligent access to a diversity of object-level sources is enabled through the knowledge-description level, the middle-tier of our three-tiered architecture depicted in Figure 1. Since legacy information systems shall be incorporated into the OM scenario without a need for modification, we propose a separate, knowledge-rich information modeling level.[2] Essentially, its purpose is to ease:

- precise selection and efficient access to information and knowledge recognized as relevant in a given task context and application situation, and
- better comprehension and interpretation by the user and the system in a given task and application context.

Every information and knowledge item[3] is described by a number of attributes for three basic dimensions of information modeling the concepts of which are defined in three corresponding ontologies (Figure 10):

1. information meta model
2. the information content
3. the creation and application context

The distinction of three different ontologies might appear a bit artificial. Indeed, sometimes objects belong to more than one ontology, e.g., employees of the company which might be authors of some documents and occur as department members in the enterprise ontology, but are also "content descriptors" of their personal files thus being elements of the domain ontology. The same holds true, e.g., for projects which can be creation context for some document thus belonging to the enterprise

---

[2] In contrast, many recent annotation-oriented projects propose to insert semantic tags into Internet documents in order to facilitate ontology-based retrieval (such as SHOE [Luke *et al.*, 1997] or Ontobroker [Fensel *et al.*, 1998]).

[3] For the ease of notation, we will call these sometimes in the following just "documents" adopting a generalized view of a document as an arbitrary knowledge container.

ontology, and content descriptor of brochures and experience reports, i.e. part of the domain ontology.

The presentation of three ontologies instead of one has *didactic reasons*, since it shows the dimensions to care about. Further reasons are the matters of *origin / creation* (e.g., the enterprise ontology is usually already formally represented somehow in a company, be it in a workflow system, or in an organization handbook, whereas the domain ontology will usually not yet exist in an explicit representation, but might partly be hidden, e.g., in the organization schema of the corporate library, or in the glossary of technical documentation) and of *stability* (usually, the information ontology is very stable, while the organization ontology may be subject to regular changes, and the domain ontology could often change, depending on the domain). Of course, these issues do also influence the questions who creates and updates the respective ontologies (or parts of a unified ontology) in a real application environment.

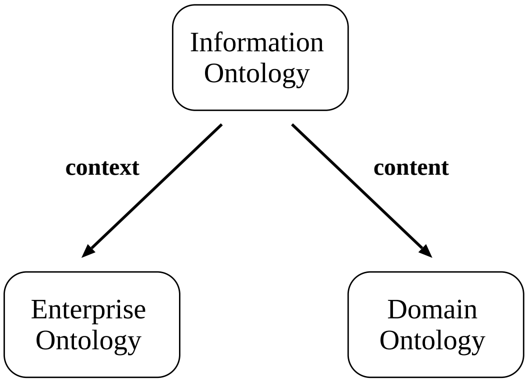


Figure 10: *Three ontologies span the basic dimensions of information modeling.*

Before discussing the question of ontology representation, we shortly sketch the respective content of our three knowledge-description ontologies.

Information meta modeling. The *information meta model* describes the different kinds of information sources with their respective structure, access, and format properties. Besides the vocabulary for such merely "syntactical" properties of specific information sources, the *information ontology* also contains generic concepts and attributes that apply to all kinds of information—like the timeliness, the author, the reliability of information, or the type of statements an information source makes; for instance, it may express *descriptive* knowledge about products and processes or *prescriptive* knowledge stating how to do certain things. Furthermore, the information ontology introduces concepts and attributes specific for certain kinds of information sources; for instance, access to an external commercial database involves costs and time delays, whereas personal competencies must be accompanied by the level of expertise of an employee and her availability.

Essentially, the information ontology (i) comprises meta-level aspects of information sources which do not directly describe the content itself, but its characteristics, (ii) it provides links into the domain ontology used for content description, and (iii) it provides links into the enterprise ontology which is used to describe the creation context and the intended utilization context of information items. The simplified example shown in Figure 11 gives an impression how the several ontologies interact.

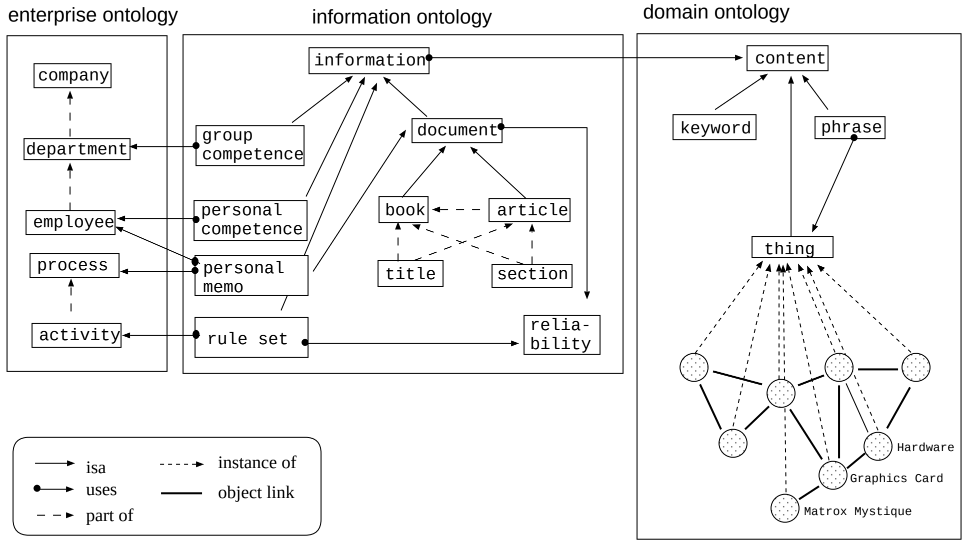


Figure 11: *Part of the knowledge-description ontologies.*

Content modeling. For modeling the content of information sources, we pursue the *conceptual indexing* [Woods, 1997] approach which is characterized by sophisticated models of the domain the documents talk about and by document indexing using pointers into these domain models (i.e., pointers to formal ontology *concepts*). This approach allows, e.g., formulation of domain-specific search heuristics [Baudin *et al.*, 1992], more precise query formulation [van Bakel *et al.*, 1996], or more comfortable knowledge browsing for the user [McGuiness, 1998]. It is a way for integrating information from different sources with different vocabulary [Kindermann and Hoppe *et al.*, 1996], and also a means for indexing non-text documents (e.g., video tapes or images) [Gordon and Domeshek, 1995]. Regarding the effectiveness of information retrieval, precision can be improved because formal models help for the disambiguation of polysemous words, and recall can be improved through ontology-based query reformulation in the case of empty or small result sets. In KnowMore, the latter approach is realized using domain-specific search heuristics for finding documents loosely or indirectly related to the search concepts [Liao *et al.*, 1999]. In [Sintek *et al.*, 2000], we discuss additional ideas for the use of domain knowledge to improve information access.

Since an OM typically contains much knowledge informally represented in texts, we link our formal ontology concepts to thesaurus-like lexical information: The concepts of the domain ontology are the basic primitives for the formal knowledge representation; attached lexical information about typical textual expressions or occurence indicators for formal concepts can help to classify [Tschaitschian *et al.*, 1997], summarize [Hovy *et al.*, 1999], and access [Guarino *et al.*, 1999] informal knowledge.

Context modeling. In addition to the usual modeling dimensions of Information Retrieval, we focus on context as highly relevant for retrieval within an organization. Context modeling concerns two issues:

- the intended application context a document is assumed to be useful for, and
- the context a document was created in, which can be exploited for an estimation of its relevance in a new, possibly similar, application context.

For instance, if a notice about some customer or supplier has been created within a certain business context—like price negotiations—this context information can be very valuable to determine the relevance of this notice in a given, new application context. For particular kinds of information, like best practice reports, lessons learned or formal design rules, the application task can be specified in advance. In our purchasing example, the business rules for buying hardware can easily be marked as mandatory to be regarded in any purchasing process. In [Abecker *et al.*, 1999b], we describe in a bit more detail another KnowMore application example—managing customer contacts for DFKI—where we experiment with two sample uses of context modeling:

1. When a company contacts DFKI for some new project, the system can search for other ongoing or past contact processes to the same company (for a research institute like ours it is not seldom that at the same time there are several independent contacts to different departments of a large international trust like Siemens or DaimlerChrysler) establishing a link to the respective

colleague, giving access to related documents and experiences, or ensuring a consistent appearance for this customer company. This service utilizes the idea that process instances are themselves also content of the OM.

2. When coming back to an activity already entered one or more times earlier in a potentially cyclic process, there are cases where the information retrieval should take into account the earlier results or decisions. For instance it might be the case that it is not appropriate to present documents which have already been seen, or to suggest decisions which were already rejected. This requires access to the dynamic context variables. It might also be possible that thoughts or notes written down in the earlier enactment of the same stage (performed by the same or some other employee) should be presented which requires that the system is aware of the detailed storage context of these documents.

In KnowMore, we describe information context in terms of static and dynamic business organization which in turn is expressed in terms of the *enterprise ontology.* Our enterprise ontology is based upon the ADONIS meta model [Karagiannis *et al.*, 1996] which is quite close to the common notions of the Workflow Management Coalition [The Workflow Management Coalition, 1996].

The KnowMore knowledge description language. The expressiveness of the representation formalism for knowledge and information modeling as well as the inferences possible over these models play a central role for the overall system functionality. From the examples above (especially the requirement of detailed domain modeling) it is already clear that at least the basic abstraction mechanisms are needed which constitute a structurally object-oriented formalism: classification of objects into classes, generalization of classes to superclasses, aggregation to express `part-of` relationships, and attribute-value assertions to specify certain class instances. We need also inferential capabilities to take into account retrieval background knowledge, for the evaluation of search heuristics, or to follow links in hypermedia documents. Further, one would like to have both a good system performance for large knowledge bases and ontologies, and a well-founded semantics based on the requirements of information modeling and retrieval (cf. [van Rijsbergen, 1989]).

Recently, some authors (cf. [Welty, 1998, Schmiedel and Volle, 1996]) noticed that there is at least no “elegant” way for easy ontology-based knowledge description with standard KR formalisms like description logics. The main problem is the representation of index concepts (e.g., `Matrox Mystique` or `graphics card`) in Figure 11. When we described non-trivial domain models for indexing purposes there appeared soon the conflicting goals that: (i) On one hand, we would like to represent index topics as concepts (i.e., *classes* of the hosting KR formalism) in order to express taxonomic relations and to have inheritance of attributes. (ii) On the other hand, we wanted to have indices as individuals (i.e., *instances* of the hosting formalism) in order to use them as attribute values for the `content` slot, and in order to have concrete attribute values and other meaningful relationships like, e.g., electric or hydraulic connections between modules in a machine-model (in concrete applications such domain-specific relations can sometimes be used for relevance propagation in the same way the `is-a` relation is used for, typically).

An underlying problem is the usually supposed semantics of subclass-superclass relationships in KR formalisms which is based on a set-oriented interpretation of classes and subset-superset relationships between these interpretations. This semantics seems not to be fully appropriate for index terms which often do not represent physical entities directly, but merely information *about* physical or abstract entities. Further, the context-specific relevance of knowledge items cannot only be propagated or inherited via the `is-a` relations between these underlying entities, but also via other relations between them.

Our current KnowMore knowledge description formalism is an object-oriented relational algebra (OCRA) which offers basic knowledge structuring and simple inferential capabilities (for more details see [Abecker *et al.*, 1998a]). One aspect of our future work might be a more integrated treatment of concepts and individuals in order to allow for better index modeling. Up to now, we have only "simulated" the aspects necessary in our sample applications. To this end, we represented index concepts as instances which can be correlated via a general type of links (the meaning of which is "has-to-do-with") which can be further specialized to more specific links (like "is more specific", or "is electrically connected"). Relevance propagation along links is done by special methods associated with each link type.

5 Processing Information Needs

In this chapter we will show how several kinds of represented knowledge interact in order to fulfill an information need. Figure 12 shows an overview of the actions performed by an information agent to answer a query. Figure 13 instantiates this scenario by a concrete example. The chosen presentation particularly separates out the several processing steps and sources of background knowledge 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.

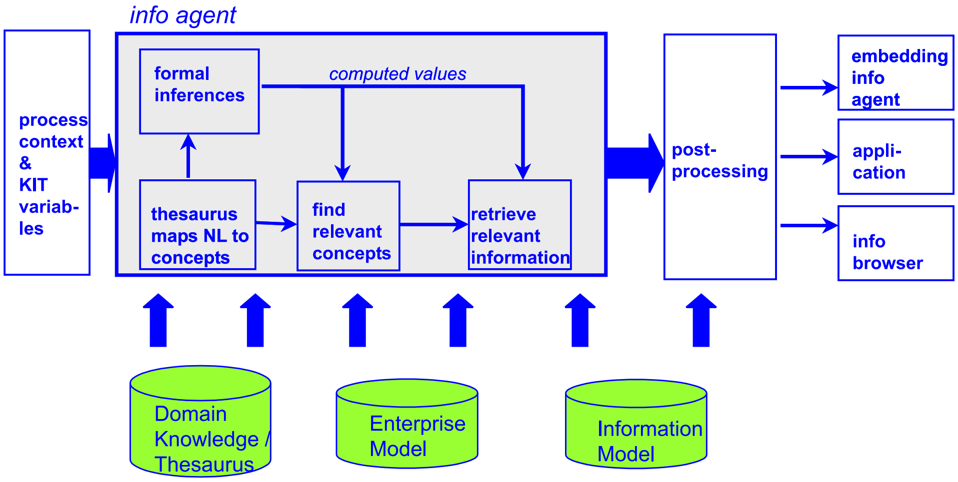


Figure 12: *Information needs are satisfied by integrated inference over formal and retrieval of informal knowledge.*

As described in Section 3, an info agent can become active in a given application situation—if the preconditions are valid—and has access to the especially created KIT variables (*which kind of product shall be purchased? what concrete product? at which price? from which supplier?*) as well as to the global and local process context (*overall business task, activity performers and their respective roles, time conditions, etc.*).

First of all, the preconditions are checked. In the KIT example of Figure 8, 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. 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. If our machinery is fully elaborated, we will also allow 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). 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 info 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.

Now, we go further into the details of the info agent's core functionality. Therefore, we have a look at Figure 13 which shows how an instantiated version of the processing schema presented in Figure 12 would be handled by the appropriate info agent. Essentially, we can identify three main processing steps:

1. Map application situation onto retrieval concepts. Since our system directly takes its query input from the application program (i.e. 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 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`).

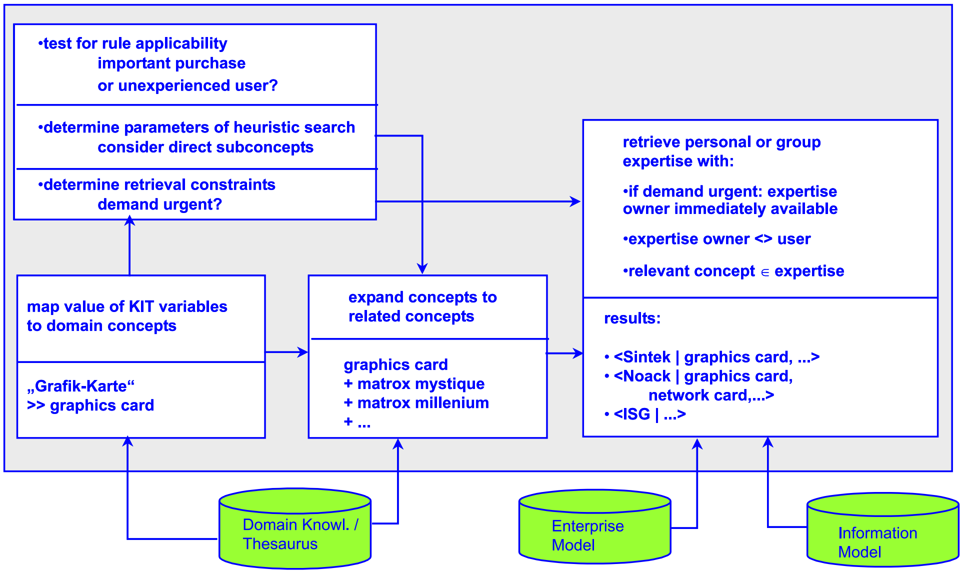


Figure 13: *The instantiated example shows how several kinds of background knowledge interact.*

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. First, 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. Then, the chosen 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.[4] 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. For such problems of vocabulary and ontology mapping, statistical techniques from thesaurus generation, as well as machine-learning methods from automatic text classification [Junker and Hoch, 1998] can be employed in the future.

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. 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.

Enriching, substituting or reformulating the query concepts is done in the second step. As already pointed out in Section 4, 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

---

[4] This last scenario is motivated by an industrial application project [DFKI GmbH, 1997] which initiated our work in automatic thesaurus construction (see [Abecker *et al.*, 1998b]).

should be utilized to support precise-content retrieval in digital libraries (see, e.g., [Welty, 1996]) and OM systems (see, e.g., [O'Leary, 1998]), or the Internet (see, e.g., [McGuiness, 1998]). 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 one about* `superconcept(x)`*."*), or rely on manual browsing through the ontology.

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. 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 how 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. On the other hand, 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.

Things get still more interesting when switching from our purchasing example to some other applications. Consider, e.g., the machine model of a complex engineering artifact as the domain of discourse used for indexing machine diagnosis experiences (see [Bernardi *et al.*, 1998c, Bernardi *et al.*, 1998a] for a fielded application of this idea). 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. 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. 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.

These examples show that simple generic search heuristics are not sufficient for complex scenarios. In [Liao *et al.*, 1999] we discuss heuristics formulation over domain ontologies as currently implemented in a prototypical personal competence search tool for our research group.

3. Retrieve information from various sources. The last step concerns retrieval in the narrower sense. At the 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. Our current retrieval machinery basically realizes some deductive database functionality.

This retrieval functionality then delivers knowledge descriptions of possibly relevant 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.

Again, we have an interaction of retrieval and formal inference, since values for *query constraints* can be formally derived, or delivered by embedded info 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 vacancy and the business trips databases. 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 to 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).

6 Implementation

Figure 14 gives an overview of the client-server architecture of our system prototype which is implemented in JAVA. The KnowMore server holds all relevant data, i.e., the business process model enriched by KIT variables and support specifications, as well as the OM archive together with the respective knowledge descriptions and the underlying ontologies. Business process models can be designed using the ADONIS commercial BPM tool (the KnowMore specific extensions are modeled as comments in the activity descriptions), and are later parsed into the KnowMore

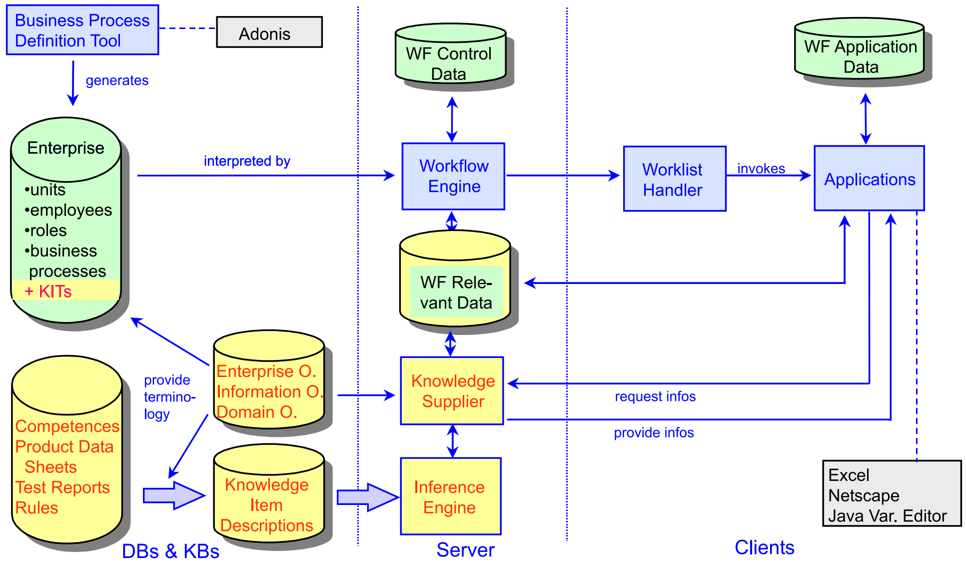


Figure 14: *The KnowMore system is implemented as a Web-enabled client-server architecture.*

representation formalism OCRA, the basic inferences of which are mapped onto conventional relational database queries (the RDBMS is coupled with JAVA via JDBC [JDBC, 1998]). In particular, classes are mapped to relations, and objects to tuples, while embedded objects are represented by their object identifiers. This mapping onto a relational DBMS provided us with innovative expressiveness at the knowledge representation level while profiting from the professional services of a state-of-the-art industrial strength database system. In the experiments already made with the system, the approach was fast enough for our purposes.[5] Of course, for future applications with huge amounts of documents to be managed and complex link structures between objects, it might be appropriate to consider the question of grounding the OCRA implementation on an object-oriented database (OODBMS).

The KnowMore server both hosts the workflow engine and the knowledge-based retrieval machinery. Workflow enactment involves two parts: the server, implemented as a JAVA application, and client worklist handlers, implemented as JAVA applets which connect to the server via standard TCP/IP sockets. The architecture and communication protocols are designed as compliant as possible with the Workflow Management Coalition standards [The Workflow Management Coalition, 1996]. So, later on, when the scenario is more stable and proved, we will consider switching

---

[5] The KnowMore OCRA is being used in the industrial application system described in [Bernardi *et al.*, 1998a, Bernardi *et al.*, 1998c].

from our “homemade” KnowMore workflow engine to a commercial one.

Besides these system core functionalities for context-aware delivery of knowledge items during workflow *enactment*, it should not be neglected that our scenario requires manifold modeling activities during process definition time. Figure 15 gives an overview of our comprehensive tool suite for facilitating these modeling efforts. The tools are described in a bit more detail in [Abecker *et al.*, 1999b]. However, some of them are still in an experimental stadium and will require further work.

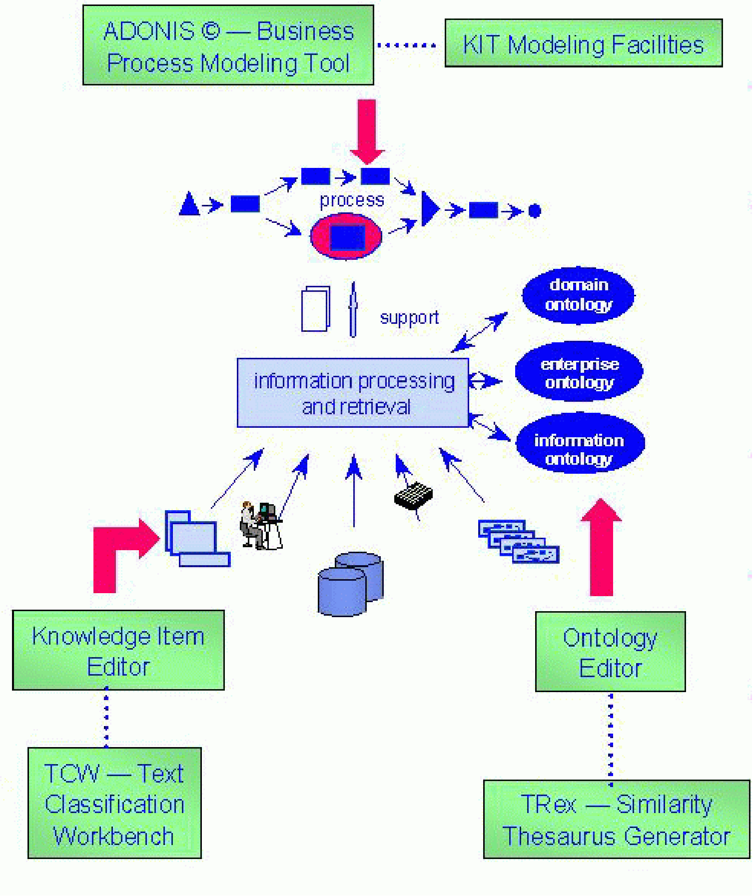


Figure 15: *A comprehensive toolbox supports the OM realization.*

7 Related & Some Future Work

Overall system approach.

The idea of coupling a user observation or task management system with a sophisticated information retrieval tool for proactive and context-sensitive information support is becoming more accepted in the last few years. [Budzik and Hammond, 2000] present an architecture for *Information Management Assistants (IMAs)* which shall observe users interacting with everyday applications and then anticipate their information needs using a model of the task at hand. The idea of assessing the user's work context for starting active delivery and for enhancing information retrieval quality is very close to our approach. The KnowMore system can be seen as an IMA in the authors' sense. Unfortunately, the ANTICIPATOR module of their system—which shall anticipate the user's future information needs on the basis of the actual work context and the stored task model—is not discussed in depth in their published work. It seems that the authors mainly build upon rather shallow, static context models which roughly determine the area of work in order to dissolve linguistic ambiguities in text retrieval. Natural-language aspects play a more important role in their work than worked-out task models as formal objects of consideration. Their WATSON system [Budzik and Hammond, 1999] observes interactions with everyday applications like word processors and web browsers, and tries to find out linguistic context of occupation. This approach is focused on the *personal* context rather than the dynamic *task* or workflow context. It would be interesting for the future to have a multiagent information assistance scenario with *task agents*—which capture the workflow context as proposed in KnowMore—cooperating with *personal agents* that hold personal activity context and user profile.

The main focus of our work is on modeling the formal task and workflow context in the sense of *just-in-time knowledge delivery* as suggested by [Cole *et al.*, 1997]: a user is always given exactly that kind of information he needs in a specific situation so that he never needs to ask for it. Systems aiming at this goal by maintaining a deep understanding of the task at hand are called *electronic performance support system.* An excellent example is the EULE2 system for knowledge-based assistance in insurance office work [Reimer *et al.*, 1998]. This system builds upon a deep, formally represented knowledge-base about all relevant concepts to be known and dealt with when working on the insurance office tasks considered.

An interesting proposal for OM-based support for running workflows, is presented by [Staab and Schnurr, 1999]. Their ideas are very close in spirit to the KnowMore approach. Compared to our system, they explore in more depth the inferential power of ontology-based retrieval on top of the Ontobroker software [Fensel *et al.*, 1998] and introduce the notion of context-based views for coupling workflow and retrieval, which is the analogue to our information needs. They build on a conventional workflow paradigm extending the well-known Petri Nets approach.

Information modeling: content and representation.

With respect to the *content* of information models (or, knowledge descriptions), the comprehensive three-dimensional approach of KnowMore seems unique in the literature. However, up to now, we have just modeled small metadata sets, which was sufficient for our demonstration examples. In order to provide a metadata set suitable for real-world applications, one needs to have a very close look into the intended application environment. We will do this in an upcoming new project funded by the European Commission (the DECOR project) which will start in summer 2000. Contributions for the design of such knowledge-description ontologies come the areas of *lessons learned archives* (see [van Heijst *et al.*, 1996] who distinguish form, content, and availability as the basic knowledge modeling dimensions) and *business knowledge navigation* (see [Steier *et al.*, 1995] who identify form, quality, and resource constraints as crucial retrieval factors besides the content). In the area of *digital libraries* the standardization of metadata sets still gains considerable interest [Dublin Core, 1997].

Regarding *representation* of information models, domain modeling languages for conceptual indexing usually provide some taxonomic and, very seldom, mereological notions (i.e. `is-a` plus `part-of` relationships). Technically, almost all systems rely on description logics (DL). We are a bit skeptical about the performance of DL-based systems in real-world applications, although, e.g., the SHOE project builds upon a high-performance DL system [Heflin *et al.*, 1999]. For information retrieval purposes, [Fuhr, 1995] showed that DL systems and deductive databases provide comparable functionalities. Our object-centered relational algebra is oriented towards the latter approach. The Ontobroker system employs F-Logic, an object-centered deductive database approach which might also be a basis for our future work [Fensel *et al.*, 1998]. Concerning the representation and interchange of ontologies and ontology-based information models, the recent developments in open Internet standards cannot be ignored for all upcoming projects (in particular cp. XMI [Object Management Group (OMG), 1998] and RDF Schema [W3C, 1999a, W3C, 1999b]).

Regarding *indexing complexity*, nowadays most systems just describe a document by one or more links to concepts of the domain ontology. Few authors proposed more detailed content models allowing, e.g., to express precoordination of concepts [van der Vet and Mars, 1996], or composite topics [Schmiedel and Volle, 1996], respectively (see [Abecker *et al.*, 1998a] for more details about different indexing approaches). The notion of index topics and the expressiveness of indexing seem to be promising areas for useful future work.

The same holds true for uncertainty handling which is often not dealt with in the area of ontology-based IR from the Internet. In KnowMore, except for some basic considerations [Wirtz, 1997], the topic was not yet in the center of our interest. However, it seems a central question to us for non-trivial scenarios, since IR problems are fundamentally characterized by usually vague user queries as well as rough document-index mappings. In traditional information retrieval, approaches for a well-founded treatment of this topic are typically based on probabilistic extensions of description logics or deductive databases [Meghini and Straccia, 1996, Rölleke and Fuhr, 1996]. Maybe promising ideas for quantitatively dealing with uncertain and fuzzy aspects of retrieving information can come from the area of Case-Based Reasoning (CBR) which is based upon the notions of similarity and utility.

8 Conclusions

In this paper, we presented the KnowMore prototype which is designed as a knowledge delivery system exploiting formal representations for automatically finding and presenting context-specific informal knowledge documents within a running workflow. From the technical point of view, it faces the following challenges:

- *Heterogeneity* of manifold possible, distributed OM sources: is handled by a uniform knowledge-description level.
- *Difficult precise-content retrieval:* is realized by an information retrieval approach on the basis of ontologies and metadata, employing search heuristics and background knowledge.
- *Active support and context-exploitation:* are enabled by declarative modeling of information needs coupled to extended workflow models via the KIT variables and the underlying domain ontology.

Parts of the software and ideas developed in KnowMore found already industrial applications or were tested in industrial case studies, in particular the OCRA object-centered relational algebra formalism and the retrieval machinery on top of it (see [Bernardi *et al.*, 1998b, Sintek *et al.*, 2000, Apostolou *et al.*, 2000]).

To realize the total KnowMore solution in a company requires considerable modeling efforts, for ontologies and knowledge descriptions as well as for extended business processes and information needs. In Section 6, we briefly mentioned the actual tool support for some of these modeling tasks. However, both tool and methodological support for these tasks must be further developed. This shall be done in the upcoming European DECOR project already mentioned above which aims at the development of pragmatic solutions for business-process oriented knowledge management. The area of ontology acquisition and maintenance—taking into account already existing knowledge sources, especially in form of text documents—requires more basic research, as well as thinking about new organisational roles for knowledge management.

Acknowledgment The work described in this paper has been supported by the German Ministry for Education and Research (bmb+f) under grant ITW 9705/3. Parts of the KnowMore project were contributed by Markus Junker, Minghong Liao, and Tino Sarodnik. We thank the anonymous reviewers for valuable hints which help to make this paper more readable and more substantial.

References

[Abecker *et al.*, 1998a] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Techniques for organizational memory information systems. DFKI Document D-98-02, DFKI GmbH, February 1998.

[Abecker *et al.*, 1998b] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Toward a technology for organizational memories. *IEEE Intelligent Systems*, May/June 1998.

[Abecker *et al.*, 1999a] A. Abecker, A. Bernardi, and M. Sintek. Enterprise information infrastructures for active, context-sensitive knowledge delivery. In *ECIS'99 - The 7th European Conference on Information Systems*, Copenhagen, Denmark, June 1999.

[Abecker *et al.*, 1999b] A. Abecker, A. Bernardi, and M. Sintek. Proactive knowledge delivery for enterprise knowledge management. In Guenther Ruhe and Frank Bomarius, editors, *Learning Software Organizations – Methodology and Applications*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

[Apostolou *et al.*, 2000] D. Apostolou, G. Mentzas, R. Young, and A. Abecker. Consolidating the product versus process controversy in knowledge management: The Know-Net approach. In *Practical Applications in Knowledge Management, PAKeM-2000*, Manchester, UK, March 2000. The Practical Application Company.

[Baudin *et al.*, 1992] C. Baudin, J. Gevins, V. Baya, and A. Mabogunje. DEDAL: using domain concepts to index engineering design information. In *Proc. of the Meeting of the Cognitive Science Society, Bloomington, Indiana*, 1992.

[Bernardi *et al.*, 1998a] A. Bernardi, K. Hinkelmann, and M. Sintek. Information systems in knowledge management: An application example. In *Practical Applications in Knowledge Management, PAKeM'98*, Blackpool, UK, March 1998. The Practical Application Company.

[Bernardi *et al.*, 1998b] A. Bernardi, K. Hinkelmann, and M. Sintek. 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.

[Bernardi *et al.*, 1998c] A. Bernardi, M. Sintek, and A. Abecker. Combining artificial intelligence, database technology, and hypermedia for intelligent fault recording. In *ISOMA-98: Sixth International Symposium on Manufacturing with Applications*, Anchorage, Alaska, USA, May 1998. TSI Press, Albuquerque, New Mexico, USA. This paper won the ISOMA-98 Best Paper Award.

[BOC GmbH, 1998] BOC GmbH. The Business Process Management Toolkit ADONIS. URL `http://www.cso.net/boc/english/index.htm`, 1998.

[Borghoff and Pareschi, 1998] U.M. Borghoff and R. Pareschi. *Information Technology for Knowledge Management.* Springer, Berlin, Heidelberg, New York, 1998. URL `http://www.iicm.edu/jucs_3_8/`.

[Budzik and Hammond, 1999] J. Budzik and K. J. Hammond. Watson: Anticipating and contextualizing information needs. In *Proceedings of the Sixty-second Annual Meeting of the American Society for Information Science.* Information Today, Inc., Medford, NJ, 1999. `http://dent.infolab.nwu.edu/infolab/papers/papersmain3.asp`.

[Budzik and Hammond, 2000] J. Budzik and K. J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of Intelligent User Interfaces 2000.* ACM Press, 2000. `http://dent.infolab.nwu.edu/infolab/papers/papersmain3.asp`.

[Cole *et al.*, 1997] K. Cole, O. Fischer, and P. Saltzman. Just-in-time knowledge delivery. *Communications of the ACM*, 40(7):49–53, 1997.

[Conklin and Weil, 1997] E. Jeffrey Conklin and William Weil. Wicked problems: Naming the pain in organizations. White Paper of Group Decision Support Systems, Inc., `http://www.gdss.com/wicked.htm`, 1997.

[Davenport *et al.*, 1996] Th. Davenport, S.L. Jarvenpaa, and M.C. Beers. Improving knowledge work processes. Sloan Management Review, Reprint Series 37(4), Summer, 1996.

[DFKI GmbH, 1997] DFKI GmbH. FAKT: Automatic Thesaurus Generation for Answering Customer Queries. URL `http://www.dfki.de/pas/f2w.cgi?daimp/FAKT-e`, 1997.

[Dieng and Vanwelkenhuysen, 1996] R. Dieng and J. Vanwelkenhuysen, editors. *Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop KAW'96, Special Track on Corporate memory and Enterprise Modeling.* November 1996.

[Dublin Core, 1997] URL `http://purl.oclc.org/metadata/dublin_core`, 1997. The Dublin Core Home Page.

[Fensel *et al.*, 1998] D. Fensel, St. Decker, M. Erdmann, and R. Studer. Ontobroker: The very high idea. In *Proc. 11th Int. Florida AI Research Symposium (FLAIRS-98)*, May 1998.

[Frei *et al.*, 1996] H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors. *Proc. of the 19th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval.* August 1996.

[Fuhr, 1995] N. Fuhr. Modelling hypermedia retrieval in datalog. In *Proceedings HIM'95.* Universitätsverlag Konstanz, April 1995.

[Gordon and Domeshek, 1995] A.S. Gordon and E.A. Domeshek. Conceptual indexing for video retrieval. In M. Maybury, editor, *Intelligent Multimedia Information Retrieval, Working notes of the IJCAI-95 Workshop, Montreal, Quebec*, August 1995.

[Guarino *et al.*, 1999] N. Guarino, C. Masolo, and G. Vetere. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3), 1999. URL `http://www.ladseb.pd.cnr.it/infor/ontology/Papers/OntologyPapers.html`.

[Heflin *et al.*, 1999] J. Heflin, J. Hendler, and S. Luke. SHOE: A knowledge representation language for internet applications. Technical Report CS-TR-4078, University of Maryland at College Park, 1999. URL `http://www.cs.umd.edu/projects/plus/SHOE/`.

[Hovy *et al.*, 1999] Eduard Hovy, Chin-Yew Lin, and Daniel Marcu. Summarist project homepage: Automated text summarization. URL `http://www.isi.edu/natural-language/projects/SUMMARIST.html`, 1999.

[JDBC, 1998] JDBC Homepage at SUN. `http://java.sun.com/products/jdbc/`, 1998.

[Junker and Hoch, 1998] M. Junker and R. Hoch. An experimental evaluation of OCR text representations for learning document classifiers. *Int. Journal on Document Analysis and Recognition*, 1(2):116–122, June 1998.

[Karagiannis *et al.*, 1996] D. Karagiannis, St. Junginger, and R. Strobl. Introduction to business process management systems concepts. In B. Scholz-Reuter and E. Stickel, editors, *Business Process Management*, Lecture Notes on Computer Science, LNCS. Springer-Verlag, 1996.

[Kindermann and Hoppe *et al.*, 1996] C. Kindermann and T. Hoppe *et al.* The MIHMA demonstrator application: bmt line. Technical report, Non Standard Logics S.A., Paris and TU Berlin, April 1996.

[Kühn and Abecker, 1998] O. Kühn and A. Abecker. Corporate memories for knowledge management in industrial practice: Prospects and challenges. In *[Borghoff and Pareschi, 1998]*, 1998.

[Liao *et al.*, 1999] M. Liao, K. Hinkelmann, A. Abecker, and M. Sintek. 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. Springer-Verlag, March 1999.

[Luke *et al.*, 1997] S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based web agents. In *Proc. of First Int'l Conf. on Autonomous Agents, AA-97*, 1997.

[McGuiness, 1998] D.L. McGuiness. Ontological issues for knowledge-enhanced search. In N. Guarino, editor, *Int'l Conference on Formal Ontology in Information Systems (FOIS'98)*. IOS Press, 1998.

[Meghini and Straccia, 1996] C. Meghini and U. Straccia. A relevance terminological logic for information retrieval. In *[Frei* et al.*, 1996]*, 1996.

[Object Management Group (OMG), 1998] Object Management Group (OMG). XML Metadata Interchange (XMI) Specification. URL `http://www-4.ibm.com/software/ad/features/xmi.html`, 1998.

[O'Leary, 1998] D. O'Leary. Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, May/June 1998.

[Prinz and Syri, 1998] W. Prinz and A. Syri. An environment for cooperative knowledge processing. In *[Borghoff and Pareschi, 1998]*, 1998.

[Reimer *et al.*, 1998] U. Reimer, A. Margelisch, B. Novotny, and Th. Vetterli. Eule2: A knowledge-based system for supporting office work. *ACM SIGGROUP Bulletin*, 19(1), 1998.

[Rölleke and Fuhr, 1996] Th. Rölleke and N. Fuhr. Retrieval of complex objects using a four-valued logic. In *[Frei* et al.*, 1996]*, 1996.

[Schmiedel and Volle, 1996] A. Schmiedel and Ph. Volle. 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. URL `http://www.info.unicaen.fr/~serge/3wia/workshop/`.

[Shum, 1998] S. Buckingham Shum. Negotiating the construction of organisational memories. In *[Borghoff and Pareschi, 1998]*, 1998.

[Simone and Divitini, 1998] C. Simone and M. Divitini. Ariadne: Supporting coordination through a flexible use of knowledge processes. In *[Borghoff and Pareschi, 1998]*, 1998.

[Sintek *et al.*, 2000] M. Sintek, B. Tschaitschian, A. Abecker, A. Bernardi, and H.-J. Müller. Using ontologies for improved information access. In *Practical Applications* *in Knowledge Management, PAKeM-2000*, Manchester, UK, March 2000. The Practical Application Company.

[Staab and Schnurr, 1999] St. Staab and H.-P. Schnurr. Knowledge and business processes: Approaching an integration. In *OM '99 - Proceedings of the international Workshop on Knowledge Management and Organizational Memory (IJCAI-99). Stockholm, Sweden*, 1999.

[Steier *et al.*, 1995] D. Steier, S.B. Huffman, and W.C. Hamscher. Meta-information for knowledge navigation and retrieval: What's in there, 1995.

[The Workflow Management Coalition, 1996] The Workflow Management Coalition. Workflow management coalition terminology & glossary. Document number WFMC-TC-10011, June 1996.

[Tschaitschian *et al.*, 1997] B. Tschaitschian, A. Abecker, and F. Schmalhofer. Information Tuning with KARAT: Capitalizing on existing documents. In *10th European Workshop on Knowledge Acquisition, Modelling, and Management*, number 1319 in LNAI. Springer Verlag, October 1997.

[van Bakel *et al.*, 1996] B. van Bakel, R.T. Boon, N.J. Mars, J. Nijhuis, E. Oltmans, and P.E. van der Vet. Condorcet annual report. Technical Report UT-KBS-96-12, Knowledge-Based Systems Group, University of Twente, September 1996.

[van der Vet and Mars, 1996] P.E. van der Vet and N.J.I. Mars. Coordination recovered. In *Informatiewetenschap 1996*. Delft, 1996.

[van Heijst *et al.*, 1996] G. van Heijst, R. van der Spek, and E. Kruizinga. Organizing corporate memories. In *[Dieng and Vanwelkenhuysen, 1996]*, 1996.

[van Heijst *et al.*, 1998] G. van Heijst, R. van der Spek, and E. Kruizinga. The lessons learned cycle. In *[Borghoff and Pareschi, 1998]*, 1998.

[van Rijsbergen, 1989] C.J. van Rijsbergen. Towards an information logic. In *Proc. of the 12th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1989.

[W3C, 1999a] World Wide Web Consortium W3C. Resource description framework (rdf) model and syntax specification. URL `http://www.w3.org/TR/PR-rdf-syntax/`, January 1999.

[W3C, 1999b] World Wide Web Consortium W3C. Resource description framework (rdf) schema specification. URL `http://www.w3.org/TR/1998/WD-rdf-schema/`, March 1999.

[Welty, 1996] C. Welty. Intelligent assistance for navigating the web. In *Proc. of the 1996 Florida AI Research Symposium*, 1996.

[Welty, 1998] C. Welty. The ontological nature of subject taxonomies. 1998. Int'l Conference on Formal Ontology in Information Systems (FOIS'98).

[Wirtz, 1997] H. Wirtz. Entwurf und Implementierung einer Beschreibungssprache für Informationsobjekte zur Unterstützung des wissensbasierten Retrievals von Unternehmenswissen. Diploma thesis, University of Kaiserslautern, August 1997.

[Woods, 1997] W.A. Woods. Conceptual indexing: A better way to organize knowledge. Technical Report Series TR-97-61, SUN Microsystems Laboratories, Palo Alto, CA, USA, April 1997.