Ontology-based Process Support

Hans-Peter Schnurr, Steffen Staab & Rudi Studer

Institute AIFB, Univ. Karlsruhe, D-76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/
{schnurr,staab,studer}@aifb.uni-karlsruhe.de

Abstract

This paper describes a principled approach towards creating an IT-support environment for knowledge workers. Starting in the analysis phase our approach paves the way for putting an intelligent assistant to work within a typical business process environment. Thereby, this support intelligence is centred around the interdependencies between documents providing the background knowledge and business processes.

Introduction

Support for the knowledge worker in her daily work may take many different guises: groupware systems and information access and retrieval tools support different knowledge processes, while workflow management covers the support for rigidly structured processes. However, what is largely missing so far is an environment that integrates the business process aspects of weakly structured knowledge work with an environment that actively supports the worker in using and adding to knowledge resources.

In addition, we must cope with three major problems that occur in practical any knowledge management setting. First, distributed knowledge should be collected and implicit knowledge should be made explicit by inferencing. Second, the knowledge worker should be able to concentrate on her core work and should not be distracted by learning new tools. Third, the knowledge worker should not be forced to work through extensive lists of information provided by an overly eager personal assistant. This means that the support provided must be concise and directly relevant.

The metaphor we have in mind is one about an intelligent assistant who looks over one's shoulder and answers questions one might have at a particular point of work. In order to avoid disconcerting dialogues, the assistant should actively propose reasonable questions and pre-fetch corresponding answers for the task at hand. Thereby, we assume that the knowledge repository is not one that is very structured, rather we presume that documents in common text processing formats are intermingled with graphical presentations, tabular information and data and knowledge bases. Hence, we claim plausibility of our approach as a real-world scenario for the information technology environment of common knowledge workers.

In this paper we want to show the design of a general methodology that paves the way for going from a common intranet and archive solution to the kind of IT support just outlined. In the following we first draft a project planning scenario that will then serve as our running example in the rest of the paper. Subsequently, we will give an outline of major business structures that we take into account, viz. intranet environments, business documents and business processes. These structures are brought together by our methodological approach.

Project Planning Scenario

Let us here consider a common project planning scenario. The setting is given as follows: a project manager in an IT-consulting firm must compile a team - mostly from current employees of his company. Thereby, he must meet the following planning requirements:

- Participants must be available for the project,
- participants should have particular technical knowledge that is needed for the project, and
- they should have some knowledge about the client in this project or at least about a client with a similar industry background.

For our scenario we have the following - realistic - assumptions: First assumption, the project manager compiles the plan with a common piece of software that supports the creation of network plans. What is important at this point is that persons who execute this task on a regular basis usually hold on to a particular tool and a particular way of executing this task with this tool. For instance, it is quite common that a project manager creates a template (or uses a template that is provided by her company) in order to execute and document the planning task.

Second assumption, the information that she relies on is drawn from (i) her personal knowledge, (ii) from the knowledge of people she asks, and (iii) from the knowledge available in other project documents and in the intranet. Naturally, only the third type of knowledge is the one that can always be accessed electronically and, thus, is the one we want to exploit for our knowledge support mechanisms. A common document containing supporting knowledge

could be a project page describing the name, the goal, the participants and techniques used in a former project.
Let us now assume that a groupware platform exists that handles scheduling tasks. The knowledge that is necessary in order to fulfill the three requirements mentioned at the beginning of this section can be found as follows:

- Availabilities may be retrieved from the scheduling database,
- technical knowledge may be found on employees' web pages,
- industry specific knowledge may be inferred from employees' participations in projects at project web pages that also name the client.

However, it is very tedious, sometimes nearly impossible, for the project manager to gather the information she needs from these different sources. The remainder of this paper will show how the knowledge for the project planning task can be provided automatically, once the project planning task has been analysed and an appropriate methodology and IT support has been introduced to the enterprise.

Ontobroker as Intranet Environment

Typical intranet environments comprise at least two technical services. First, they offer means to store documents. We here may assume that all documents are available in the intranet. Second, intranet environments offer technology for navigating the intranet environment and finding information. Our KM methodology builds on the framework given through the Ontobroker approach as described in [Benjamins et al., 1998] and [Decker et al., 1999]. The main components of Ontobroker from a functional view point are the underlying ontology, the annotated document sources and the query and inference engine.

Ontology

The ontology is a specification of a shared conceptualization [Gruber 1993]. In our application the ontology provides the semantic basis on which we build for inferences as well as for accessing the facts. It comprises three different types of assertions (cf. Table 1).
First, a set of concepts that one talks about. Second, a set of attributes that link different concepts and also associate concepts with properties. Third, a set of rules that define the semantics of concepts and attributes. These rules are used to enforce a particular semantics, e.g. the symmetry of relations.

Annotated Document Sources

Ontobroker uses the ontology as a conceptual basis for concepts and relations it can handle and reason about. However, only the annotated document sources establish the linkage between abstract concepts and real facts. Hence, it may be described in the ontology what a project is in general, but a particular project, call it ''OntoIce'', must be declared to be a project on the corresponding project page. Ontobroker allows for several types of declarations: an extension to HTML, XML and RDF. We here consider the XML variant, i.e. a particular set of XML-tags linking text instances, e.g. projectName, to concept and attribute definitions in the corresponding ontology. For instance, ''OntoIce'' is defined as the name of a project (in Table 2).

Query and Inference Engine

A gatherer searches through the intranet documents, extracts all the facts stated in these document, e.g. that a project named ''OntoIce'' exists and that it has the member ''S. Decker'', and stores these facts in the knowledge base.
The query and inference engine then allows the retrieval of these facts from the knowledge base. Thereby, Ontobroker combines the advantages of (thesaurus-supported) keyword-based search methods with database queries. In addition, it allows to infer implicit knowledge.

Business Documents as Knowledge Repository

Current business documents come in many different forms. It is quite common that these different forms are often standardized up to a certain point, indeed they often come with a particular semantics, such as the short notes that come with check boxes like "please answer" or "urgent". Similarly, letters are usually composed with a particular corporate identity in mind.

Table 1: Partial ontology for human resource management.

Concepts	Attributes	Rules
Object[].	Person [firstName =>> String;	FORALL Proj1, Pers1
Person :: Object.	lastName =>> String;	Proj1 : Project[member ->> Pers1]
Employee :: Person.	email =>> String;	<->
Manager :: Employee.	phone =>> String;	Pers1 : Person[participantOf ->> Proj1]
Consultant :: Employee.	participantOf =>> Project;	
Project :: Object.	hasCompExperience=>>Company]	FORALL Pers1, Proj1, Comp1
Company :: Object.		Proj1 : Project
Manufacturer :: Company.	Project[projectName =>> String;	[member ->> Pers1,
FinanceCompany :: Company.	projectGoal =>> String;	client ->> Comp1]
Insurer :: FinanceCompany.	client =>> Company;	->
LifeInsurer :: Insurer.	member =>> Person;	Pers1 : Person
Bank :: FinanceCompany.	leader =>> Person]	[hasCompExperience->> Comp1]

Table 2: The XML project page.

```
<project>
    <projectName>OntoIce</projectName>
<projectGoal>Bringing Knowledge Management to
<client type=„LifeInsurer“>Nordic Life</client>, Spitzbergen
</projectGoal>
...
The task of our team,
<member>S. Decker</member>,
...
</project>
```

With our approach we go even one step further, since we also link these documents to the enterprise's ontology (or ontologies). This leads us to annotated document templates, and, thus, makes them available as an explicit knowledge repository. Beforehand, however, we want to introduce the technical platform that we rely on.

SGML/XML Documents as Technical Platform

SGML (Standard Generalized Markup Language) and a subset of it, XML (eXtensible Markup Language) [Bray et al, 1998], are standardization efforts that aim at a general scheme for exchanging documents. In our scenario, SGML/XML gives us the power to reason about document structures and contents as will be described further down in the paper.

Annotated Document Templates

Based on the considerations from the preceding sections, we have decided to build our approach on annotated document templates. Annotated document templates outline the general structure of a business document. For instance the XML-tags in Table 2 – i.e. without the actual facts - might serve as a template for project descriptions in general. An automatic compilation from ontologies to DTDs establishes a communication layer between documents and ontological annotations [Erdmann & Studer, 1999]. When the user fills in parts of the document in order to complete his business task, then she connects the information she provides with corresponding metainformation.

In spite of this structuring, the templates leave a lot of leeway for user adaptations that may become necessary to describe the plan any further, e.g. with text, tables or even figures.

Business Processes

In this section, we consider some related work on business processes with a (comparatively) rigid structure that we build upon. This work adopts a formal high-level Petri net point of view. Particularly interesting for our approach is the use of document contents for controlling the business process in a special version of Petri nets, viz. so-called SGML nets.

Rigid Business Processes and SGML nets

An SGML net (cf. [Weitz, 1998]) comprises the basic concepts of Petri nets, but instead of just a flow of atomic markers it also allows the modeling of a flow of SGML documents: the transition rules perform operations on the SGML documents, such as checking off a text passage or iterating over a list. These kinds of transition rules are defined by way of match operators that make use of the structure provided by SGML.

Still, the way SGML nets are defined only captures the (comparatively) rigid parts of the workflow. From the system's point of view the less rigid parts are equivalent to transitions with non-deterministic, since exogeneously triggered, operations. The selection of an appropriate transition operation is delegated to the user. In order to support her decision making, we want to present appropriate knowledge from the document repository of the enterprise. This purpose is realized by context-based views that use pattern-matching on XML-structures.

Context-based Views

The observation that we build on is that knowledge work, though often unstructured, still involves a large amount of subtasks and document parts the structure of which is stable over time. We analyse these structures with the goal of finding the interdependencies between the "process"-oriented subtasks and the "document part"-based representation. The underlying idea is that the documents themselves describe the progress that has been made towards achieving the goal and, thus, indicate the knowledge that may be of help for the worker - similarly as SGML structures determine the transitions that may occur at a given state of the system.

For example, during the task of compiling a team the worker wants to get some support with finding the appropriate team members. The current work document(s) represent the task that must be performed, e.g. a project plan needs to be compiled. This work involves communicating with prospective project participants. The problem often lies in identifying appropriate participants. Our methodology allows the establishment of blueprint questions like ''Which person in the company knows about X and has capacity for projects as of Y and what is her phone number?

Integrating Context-Based Views and SGML nets

In our model, the state of the system is given by the combination of a state variable (corresponding to places in the SGML net) and implicitly through the way SGML queries match certain semantic predicates. Depending on this state, the system may either initiate a new task (a transition in the SGML net) or it may display context information, like the fax number of an addressee.

The kind of help that could be useful is defined by the state that the document is in and by the task which is executed on it. Thereby, tasks and states are interrelated. The states determine whether a task may be executed and the executed tasks (together with the intitial states) determine the state of the document. It is common to annotate transitions with logic predicates that define when a transition may be executed. In addition, we define logic predicates at transitions and at places that define context-based views.
Rules describe what help to offer depending on the facts that are stated in the document and the state that is currently reached by the document. By this way a flexible, declarative mechanism is exploited that might be easier to maintain than hand-crafted imperative code.

Methodology

With our methodology we approach an integration of the knowledge management and business process aspects described in the preceding sections. As is widespread practice, we distinguish the phases of building and using the IT support environment (cf. Figure 1). Furthermore, we divide the first phase - as is also common - into analysis and design.

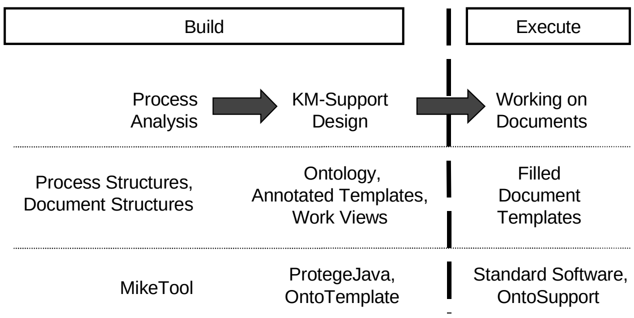

Figure 1: The methodological approach.

During the knowledge management preparation phase (build), existing business structures, i.e. processes and documents, are analysed. The results are used to design the knowledge management support structures. During the execution phase these structures are given life and, by the way of everyday work, the knowledge repository fills up. These three phases are now elaborated in this section.

Process and Document Analysis

In this first step of the build phase, we focus the analysis on processes which should be supported by the Knowledge Management system and on documents, playing a key role in the business process and the knowledge process. Keeping in mind that weakly structured processes are in our scope, a modeling approach such as applied in common workflow management systems is too rigid and therefore not applicable to a knowledge management setting. Nevertheless, our assumption is, that knowledge work, though often unstructured, still involves a large amount of subtasks parts of which are stable over time.
Our approach employs a division of processes into tasks and subtasks. The context is analysed and typical questions one might have at a particular point of work are compiled. Thus, one may find the interdependencies between the subtasks and the document-part-based representation. The analysis of these subtask and document structures is supported with the MIKE tool (Model-based and Incremental Knowledge Engineering) [Angele et al., 1998] [Steinberg, 1999]. MIKE integrates semiformal and formal specification techniques together with prototyping into a coherent framework. For the semiformal representation we use a hypermedia-like formalism which serves as a communication basis between knowledge worker and knowledge engineer during knowledge acquisition.

KM-Support Design

In the second step of the build phase, we design the Knowledge Management support system. This support system contains a domain ontology describing the content of the documents, a context-model with an overview of context-based views and samples of annotated templates (cf. the basic dimensions of ontological modeling: enterprise, information and domain ontology in [Abecker et al., 1998]). These knowledge descriptions are generated according to the task and document structures having been defined in the process analysis phase. The transition of the specification defined with the MIKE tool in the first step of the build phase into annotated templates, context-based views and domain ontologies in the second build phase is supported with ProtegeJava [Eriksson et al., 1995] and OntoTemplate, a tool to design templates. ProtegeJava is a suite of tools to model a domain. It contains a graphical editor to model ontologies. The process and the document structures acquired with the MIKE tool in the process analysis phase, constitutes the input to ProtegeJava. By this way one may smoothly integrate existing ontologies with the new domain model or integrate the new domain model into upper ontologies.

Execution Phase

As already indicated people use annotated templates that are then filled during the course of regular work. The user may add further annotations if she wants to, but is not obliged to do so. The environment is embedded in common standard software that may become even more accessible for the programmer of such environments, since XML has gained widespread popularity among major suppliers of standard office software. The knowledge management environment we provide is geared towards natural and convient use by the knowledge worker. In particular, this environment realizes one of the major ideas of knowledge management, viz. that an IT tool may only act as a facilitator for sharing, creating or retrieving knowledge, but never as a key player in creating, evaluating or contributing knowledge. Hence, the use of our support environment should never be made obligatory. Rather, the intention is the provision of active help for the user by push technologies - without annoying the user.

Related Works

Benjamins et al., 1998] already outlined how Ontobroker could be used for knowledge management, however in their approach the user had to bear all the burden of doing the right things at the right time, while our approach goes in the direction of assisting the user with pieces of knowledge that might be useful for him in his very next task.

Our considerations also fit well with ones by [Abecker et al., 1998] about the nature of processes and document repositories. However, with our approach we breathed life into these formerly rather abstract notions.

Nearest to our integration of workflow and knowledge management aspects are works by [Huber, 1998], [Reimer et al., 1998], and [Ackerman & Mandel, 1999]. [Huber, 1998] builds on a Lotus Notes intranet environment that lets the user define a simple ontology and small workflows. However, his tool cannot query facts, not to speak of implicit knowledge, but only documents.

[Reimer et al., 1998] supports the user with particular tasks. For this purpose, they use rather rigid process structures that are build from declarative business rules. We, in contrast, leave all the decision with the user and try to provide him with information that might facilitate his problem solving.

[Ackerman & Mandel, 1999] describes an approach that hierarchically structures tasks and abstracts from different types of data collections in order to support the users in their purpose of analysing astronomical data. Thus, they pursue a goal that is comparable to ours. However, their application does not carry over to the fine level of knowledge structures *inside* of documents. With our approach we intend to reach a higher degree of flexibility as far as representational level and task goals are concerned.

Conclusion

We presented an approach towards supporting the knowledge workers in an enterprise that is based on the tight interconnection of documents, knowledge, organization and processes. In order to make the system practicable, we circumvent the knowledge annotation bottleneck by providing templates that are filled anyway in the course of the work process. In order to succeed, however, we plan to support the user in his own setting (i.e., Framemaker ™, Lotus Notes ™, Word ™ , etc) and though this will take some pain to achieve it will be a must for successful KM technology.

Our approach still shows several loose ends that need to be tied up by further research. We only mention here mappings between ontologies and XML structures (DTDs), and the formal basis of the integration of workflow aspects with knowledge management.

Acknowledgments. The research presented in this paper has been partially funded by the German Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) under grant number 01IN802 (project ``GETESS´´).

References

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

M. Ackerman and E. Mandel. Memory in the Small: Combining Collective Memory and Task Support for a Scientific Community. *Journal of Organizational Computing and Electronic Commerce*, accepted for publication in 1999.

J. Angele, D. Fensel, D. Landes, & R. Studer. Developing Knowledge-Based Systems with MIKE. *Journal of Automated Software Engineering*, 5(4), 1998, 389-418.

V. R. Benjamins, D. Fensel, & A. Gómez Pérez. Knowledge Management through Ontologies. In: *PAKM-98*, pp. 5.1-5.12, 1998**.**

T. Bray, J. Paoli, C.M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0.1998, http://www.w3org/TR/REC-xml

S. Decker, M. Erdmann, D. Fensel, & R. Studer. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In R. Meersman et al. (eds.): *Database Semantics, Semantic Issues in Multimedia Systems*, pp. 351-369, Boston, MA: Kluwer, 1999.

M. Erdmann, R. Studer. Ontologies as Conceptual Models for XML Documents. Institute AIFB, Research Report, 1999.

H. Eriksson, Y. Shahar, S. W. Tu, A. R. Puerta, and M. Musen. Task Modeling with Reusable Problem-Solving Methods, *Artificial Intelligence*, 79(2):293---326, 1995

T.R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), 1993.

H. Huber. Document Research Based on Collaborative Provided Structure Knowledge. *PAKM-98*, pp. 11.1-11.9, 1998.

M. Kifer, G. Lausen, & J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42, 1995.

PAKM-98: U. Reimer (ed*.), Proceedings of the 2nd International Conference on Practical Aspects of Knowledge Management*, Basel, Switzerland, October 29-30, 1998.

U. Reimer, A. Margelisch, B. Novotny, T. Vetterli. EULE2: A Knowledge-based System for Supporting Office Work. *ACM SIGGROUP Bulletin*, 19(1), pp. 56-61, 1998.

U. Steinberg. MIKE-Tool II : Ein CAKE-Werkzeug. Master's thesis. AIFB, Univ. Karlsruhe, 1999 (in German).

Wil M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems and Computers*, 8(1):21--66, 1998.

W. Weitz. Combining Structured Documents with High-level Petri-nets for Workflow Modeling in Internet-based Commerce. *Journal of Cooperative Information Systems*, 7(4), 275-296, 1998.