Knowledge Processes and Ontologies

Steffen Staab and Rudi Studer, *Institute AIFB—University of Karlsruhe and Ontoprise GmbH*
Hans-Peter Schnurr, *Ontoprise GmbH*
York Sure, *Institute AIFB—University of Karlsruhe*

This approach for ontology-based knowledge management includes a tool suite and methodology for developing ontology-based KM systems. It builds on the distinction between knowledge process and knowledge metaprocess and is illustrated by CHAR, a knowledge management system for corporate history analysis.

Increases in product complexity, the move toward globalization, the emergence of virtual organizations, and the increase in focus on customer orientation all demand a more thorough and systematic approach to managing knowledge. Knowledge management is a major issue for human resource management, enterprise organization, and enterprise culture. But IT plays a major supporting role in managing knowledge.

You typically build IT-supported KM solutions around an organizational structure that integrates informal, semiformal, and formal knowledge to facilitate its access, sharing, and reuse.[1] In such contexts, where knowledge has to be modeled, structured, and interlinked, ontologies can help formalize the knowledge shared by a group of people.[2]

In this article, we present an approach for ontology-based KM that includes a suite of ontology-based tools as well as a methodology for developing ontology-based KM systems. Our approach, shown in Figure 1, builds on the distinction between *knowledge process* (handling knowledge items) and *knowledge metaprocess* (introducing and maintaining KM systems).

Ontologies constitute the glue that binds knowledge subprocesses together. Ontologies open the way to move from a document-oriented view of KM to a content-oriented view, where knowledge items are interlinked, combined, and used. The method for developing KM systems that we outline in this article (that is, the knowledge metaprocess) extends and improves the CommonKADS method[3] by introducing specific guidelines for developing and maintaining ontologies.

Our approach shows that you can clearly identify and handle different subprocesses that drive the development and use of KM applications. You support these subprocesses by appropriate tools that are tied together by the ontology infrastructure.[4]

Knowledge items versus documents

Companies have typically pursued a very simple, pragmatic approach for introducing KM into the enterprise (moving along the knowledge metaprocess in the yellow circle in Figure 1), which means that they have picked only the low-hanging fruit. We summarize this approach in the left column of Figure 2. What appears preeminent in this approach is the focus on handling documents (steps 2 and 3) and the existing but minor role of the appendix "process."

In spite of its immediate successes, this approach presents several disadvantages. In particular, it often leads to the consequence that the knowledge process steps (the blue circle) of creation, import, captur-

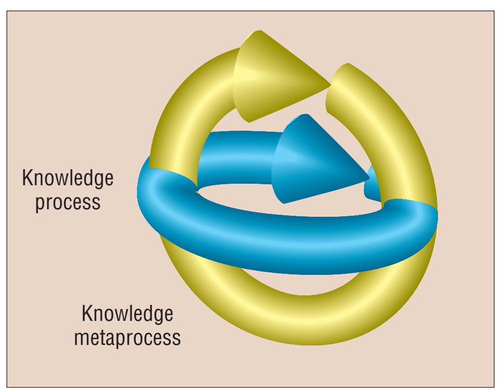


Figure 1. Two orthogonal processes with feedback loops illustrate our approach to knowledge management.

1094-7167/01/$10.00 © 2001 IEEE

ing, retrieving and accessing, and using are only very loosely connected, if at all. The underlying reason is that for each of these steps different types of business documents play a major role, which makes “knowledge re-use”—let alone knowledge refinding—extremely difficult .

Relevant knowledge items can appear in a multitude of different document formats: text documents, spreadsheets, presentation slides, database entries, Web pages, construction drawings, or email, to name but a few. The challenge lies in how you handle the knowledge.

At the one extreme, in traditional document management, IT support for KM cannot take advantage of the document contents themselves but only their explicit or implicit classification. At the other extreme, there are expert systems that structure and codify all knowledge in the system. However, although such an approach might sometimes be appropriate, not everything can be codified. A lot of knowledge is created sporadically, and the value of its reuse can only be demonstrated over time. You must therefore search for an adequate balance between reuse, level of formality, and cost. For instance, certain help desk scenarios imply long-term use of extremely well-defined knowledge items.[5] In these cases, it might be economically advantageous to spend some time and money on coding. On the other hand, a hallway discussion is usually not worth codifying at all, because it might not be reusable.

To balance conflicting needs and manage various degrees of encoded knowledge, it can be useful to employ the concept of metadata, which we can define in two complementary ways:

- *Data describing issues related to the content of data.* We divide this category into two orthogonal dimensions: the formality of the data and the containment of the metadata. In the first dimension, metadata might range from very informal descriptions of documents (like free-text summaries of books) to very formal descriptions (like ontology-based document annotation). In the second dimension, parts of metadata might be internal to the data that is described (like an HTML author tag) while others might be stored completely independently from the document they describe (like a bibliography database that classifies the documents it refers to but does not contain them).

	Document focus	Knowledge item focus
1.	Find out what the core knowledge needs are.	
2.	Find out which *business documents* and *databases* deal with these knowledge needs.	Find out which *knowledge items* deal with these knowledge needs.
3.	Build an *infrastructure* for your organizational memory system.	Find out which *knowledge processes* to allow for creation, handling, and process support of and around knowledge items.
4.	Re-organize *processes* to deal with creation and distribution of knowledge.	Build an *infrastructure* for your organizational memory system.

Figure 2. The document focus compared to the knowledge item focus.

- *Data that describes the structure of data.* In our case, you can call this type of metadata “meta metadata” because it describes the structure of metadata. This distinction boils down to an ontology that formally describes the domain of the KM application, possibly including parts of the organization and the information structures.[6]

Metadata can help condense and codify knowledge for reuse in other steps of the KM process. It can also help link knowledge items of various degrees of formality together, thus allowing a sliding balance between depth of coding and costs.

Knowledge process

Once you fully implement a KM system, knowledge processes essentially revolve around the following steps (see Figure 3):

- *Creation or import.* The contents need to be created or converted so that they fit the conventions of the company.
- *Capture.* Knowledge items have to be captured in order to determine their importance and how they mesh with the company’s vocabulary conventions.
- *Retrieval and access.* This step satisfies the searches and queries for knowledge by the knowledge worker.
- *Use.* The knowledge worker will not only

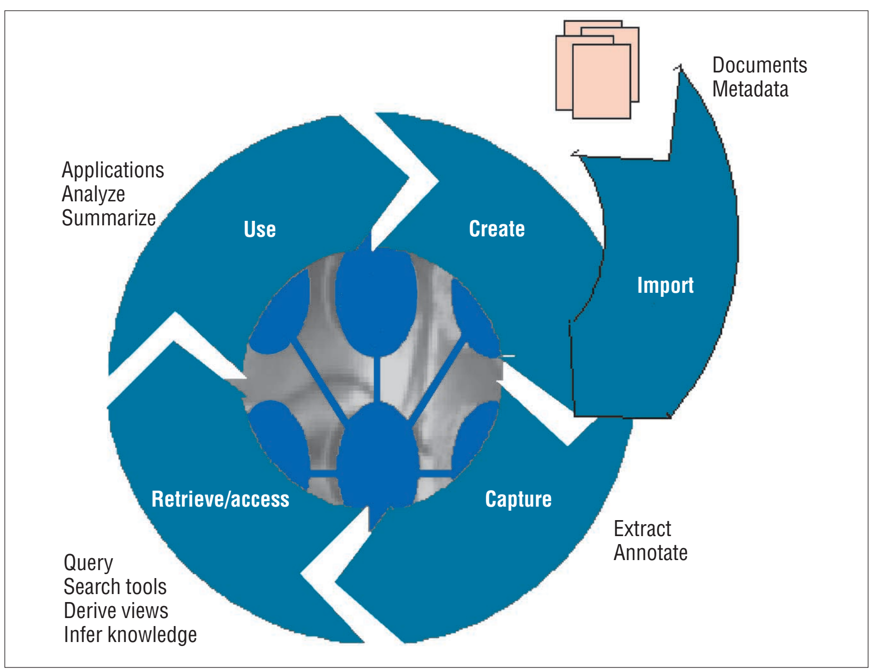


Figure 3. The knowledge process.

Table 1. Degrees of formal and informal knowledge.

Degree	Model	Interface	Example
Thoroughly formal	Relational	Form interface	Database interface
Formal	Content-structured document	Tight XML structure	XML-EDI
Partially formal	Document template	Loose XML structure	Investment recommendation template
Informal	Free text	No predefined structure	ASCII text file

recall knowledge items, but will process them for further use.

Knowledge creation

Creation of computer-accessible knowledge typically moves between the two extremes of formal and informal. Comparatively deep codification can often be done without requiring any extra effort. Business documents, in general, do not arbitrarily change but often come with some inherent structure, like engineering requirements for quality management. Our idea is to embed the structure of knowledge items into *document templates*, which are then filled on the fly by doing daily work.[7]

The granularity of this knowledge then lies in the middle between the extremes of coarsely representing business documents and representing them too finely. As Table 1 illustrates, there are several degrees of formality between formal and informal knowledge.

In Table 1, we use the term *content-structured documents* to refer to XML structures that are tightly (sometimes explicitly, sometimes implicitly) linked to a domain model. For instance, XML-EDI documents come with a predefined structure alluding to a standard framework for exchanging data, such as invoices, healthcare claims, or project status reports. By the term *document templates*, we mean similar structures that come with a larger degree of freedom, including large chunks of informal knowledge items such as the one found in Figure 4.

Careful analysis of the in-use knowledge items lets you add formal knowledge parts into the process of creating documents. Doing so pushes the degree of formality slightly upwards without endangering overall system use.

Knowledge import

For many KM purposes, importing knowledge items into the KM system has the same or more importance than creating them. You can liken the situation to data warehousing, except that the input structures are more varied and the target structures are much richer and more complex.

For imported knowledge, accurate access to relevant items plays an even more important role than for homemade knowledge. For homemade knowledge items, people might act as a backup index, but they can't for recently imported knowledge that no one has yet seen. In fact, studies have shown that the parts that cover imported knowledge are less heavily exploited than those covering homegrown ones.[8]

```
<investmentrecommendation>
<author>                                   Henrik Oppermann          </author>
<plandate>                                 October 18th, 2003        </plandate>
<interviewpartners>
                     <name>                     York Sure             </name>
                     <name>                     Hans-Peter Schnurr    </name>
                     <name>                     Steffen Staab         </name>
</interviewpartners>
<recommend>                                     strong buy            </recommend>
<details>
                     <peergroup>                …                     </peergroup>
                     <…>                        …                     <…>
</details>
</investmentrecommendation>
```

Figure 4. An example of a knowledge item, in this case an investment recommendation template.

Knowledge capture

Once you create knowledge items, the next step is to capture their essential contents. There are several indexing and abstracting techniques typical of library science. In addition, we provide a means to capture document excerpts as well as interlinkage between excerpts by our tool, OntoAnnotate.[4]

OntoAnnotate, illustrated in Figure 5, lets you create objects and describe them with their attributes and relations. Describing objects this way helps exploit knowledge found on Web pages, in spreadsheets, or in text documents. In the example shown in Figure 5, OntoAnnotate captures several facts, such as company M.A. Hanna selling its Shape Distribution Business to GE.

Through this annotation process, you create metadata that conform to the ontology and that can be aligned with related information to yield analyses and derivations like those we describe in this article. The origins of the metadata may be used to validate the overall information.

Knowledge retrieval and access

You typically perform large parts of knowledge retrieval and access through a conventional GUI. You can use the ontology to derive further views of the knowledge. In particular, you can exploit the ontology for navigation purposes, offering navigation structures the way Yahoo does. Knowledge workers can explore what is in the organization's collective memory without being required to ask a particular question—which is often a hard task for newcomers.

Also, using an ontology lets you derive additional links and descriptions. For example, the ontology can help you derive state descriptions for points in time for which no explicit data exists (such as the current structure of a company that has changed its organization by mergers or acquisitions). Thus, it can provide new hyperlinks that aren't explicitly given. Ontologies can help complete views of your captured knowledge without requiring that all information is actually given.

Knowledge use

Knowledge use, which deals with the most intricate points of KM, is the part that is most often neglected. Many KM systems assume that once some relevant document is found, everything is done. Eventually, however, the way to use knowledge from the organization's collective memory becomes quite involved. Topics such as proactive access, personalization, and in particular, tight integration with subsequent applications play a crucial role for the effective reuse of knowledge. Very often it is not even the knowledge itself that is of most interest, but the derivations that can be made from the knowledge. For instance, in our case study, no single knowledge items about a company might be relevant to a market analyst, but the overall picture presented by the analysis can be quite relevant.

In addition, usage data tells a lot about the organization's memory and about the organization itself. You can analyze which processes, customers, and techniques are tied to core figures of the organization. In one of our cases, we identified the important role of a sales representative in South America only by his pervasive knowledge stored in the organizational memory about this market. Without the representative, the South American market might have gotten lost and, hence, the company began initiatives to avoid this situation.

Knowledge metaprocess

Over more than a decade, CommonKADS has been generally successful as a methodology for developing KM systems.[3] It took until the 1990s for research to begin identifying the virtues of ontologies for sharing and reusing knowledge. But there is little work that tightly integrates ontology development into an overarching methodology for introducing knowledge management systems, such as CommonKADS. In contrast to seminal related work in the area of ontology development,[2,9] our focus lies on the application-oriented development of ontologies, including feedback and requirements from the KM application.

Our model for ontology development ranges from the early stages of setting up a KM project to the final rollout of the ontology-based KM application. In this article, we integrate some lessons learned from our experiences into the steps to be taken to perform ontology development activities. Figure 6 shows the ontology development process.

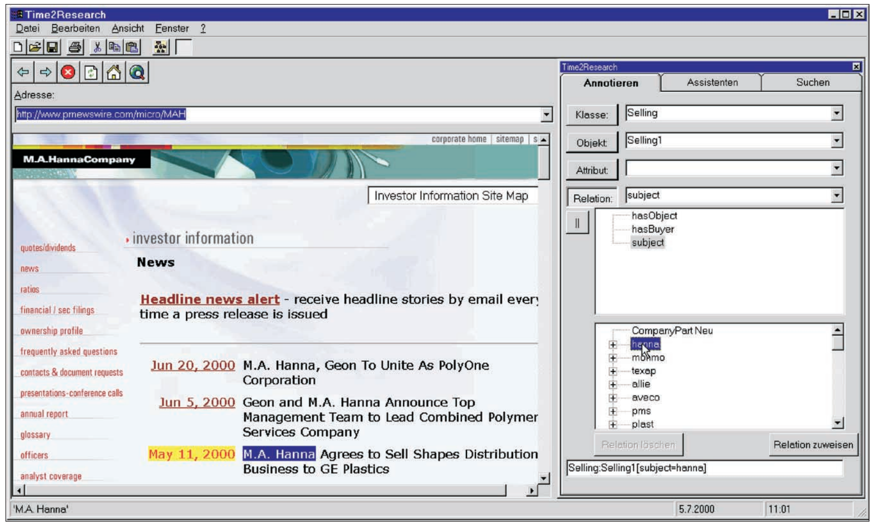


Figure 5. How OntoAnnotate works when capturing knowledge about a sale.

Feasibility study

Any KM system can only function satisfactorily if it is properly integrated. Several factors other than technology can determine success or failure. To analyze these factors, you have to perform a feasibility study to identify problem or opportunity areas and potential solutions. You also perform a feasibility study to put these problems or opportunities into a wider organizational perspective.

The feasibility study can help you determine economical and technical project feasibility. It can help you select the most promising focus area and the best solution to any potential problems. For our purposes, we adopted the kind of feasibility study described in the CommonKADS methodology. The feasibility study should be carried out before actually developing ontologies because it serves as a basis for the kickoff phase.

Kickoff phase

The kickoff phase's output product is an ontology requirements specification document (see Figure 7). The kickoff phase describes what an ontology should support and sketches the planned area of the ontology application. It should also guide an ontology engineer to decide about inclusion, exclusion, and the hierarchical structure of concepts in the ontology. In this early stage, you should look for already developed and potentially reusable ontologies.

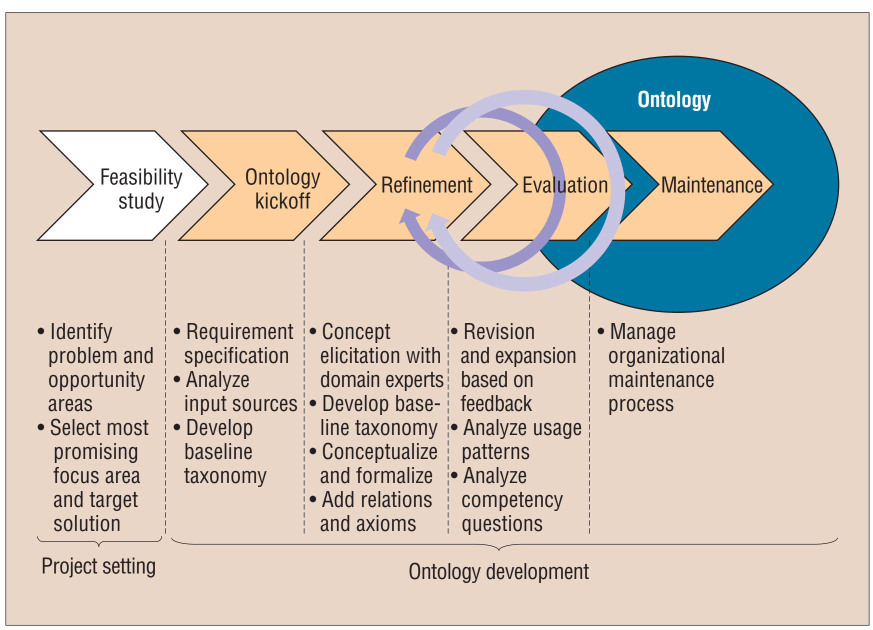


Figure 6. The ontology development process or knowledge metaprocess.

Domain: Business strategy in the chemical industry
Date: 2000/11/26
Ontology Engineer: T. Model

Goal of the ontology:
- Tracking and analyzing corporate business histories

Domain and Scope:
- Mergers & acquisitions, restructurings, management changes, and other strategic activities in the chemical industry

Supported applications:
- Web-based corporate history analyzer

Knowledge sources:
- Research analysts (domain experts)
- Related Web sites (company homepages, chemical industry networks)
- Newspaper articles
- Ad hoc news tickers

Users and use cases:
- Users: Research analysts, strategic consultants
- Use Case 1: Track strategies of specific companies
- Use Case 2: Analyze strategic moves of competitors

Competency questions:
- Attached competency questionnaire

Potentially reusable ontologies:
- Not known

Figure 7. Ontology requirements specification document.

In summary, the kickoff phase should clearly detail several pieces of information:

- the ontology's goal,
- its domain and scope,
- the applications it supports,
- its knowledge sources (like domain experts, organization charts, business plans, dictionaries, index lists, or db-schemas), and
- its potential users and usage scenarios.

It should also include a competency questionnaire like the one shown in Table 2—essentially an overview of possible queries to the system that can indicate the scope and content of the domain ontology. Finally, the kickoff phase should detail any potentially reusable ontologies.

Refinement phase

The goal of the refinement phase is to produce a mature and application-oriented target ontology according to the specification given by the kickoff phase. We divide this phase into different subphases:

- Gathering an informal baseline taxonomy containing relevant concepts given during the kickoff phase.
- Eliciting knowledge from domain experts based on the initial input from the baseline taxonomy to develop a seed ontology that contains relevant concepts and describes the relationships between them. The seed ontology is expressed at an epistemological level.
- Transferring the seed ontology into the *target ontology* expressed in formal representation languages like Frame Logic, Description Logic, or Conceptual Graphs.

Using potentially reusable ontologies (identified during the kickoff phase) can improve the speed and quality of the development during the whole refinement phase. These ontologies might provide useful hints for modeling decisions.

Evaluation phase

The evaluation phase serves as a proof of the usefulness of developed ontologies and their associated software environment. In this step, the ontology engineer checks whether the target ontology satisfies the ontology requirements specification document and whether the ontology supports or answers the competency questions analyzed in the kickoff phase of the project.

In this step you also test the ontology in the target application environment. Feedback from beta users might be valuable input for further refinement of the ontology. The prototype system should track the ways users navigate or search for concepts and relations. You can then trace what areas of the ontology are used most often.

This phase is closely linked to the refinement phase. An ontology engineer might need to perform several cycles until the target ontology reaches the specified level. Rolling out the target ontology finishes the evaluation phase.

Maintenance phase

Specifications for ontologies often need to change to reflect changes in the real world. To reflect these changes, ontologies have to be maintained frequently. Maintaining ontologies is primarily an organizational process. There must be strict rules for the update-delete-insert processes within ontologies.

We recommend you gather changes to the ontology and initiate the switch-over to a new version of the ontology after thoroughly testing possible effects to the application. As in the refinement phase, feedback from users might be valuable for identifying changes.

Knowledge metaprocess instantiated

Actively tracking and managing relevant knowledge is a major task for knowledge-intensive companies. While the correct analysis of market situations and competitors are critical requirements for success, the failure to provide adequate knowledge about one's business environment can invite failure.

Feasibility study

In the feasibility study we found that management and professionals have a hard time gathering information, analyzing it, and performing their operational work. The task of the corporate researcher is to track relevant knowledge and communicate it to stakeholders within the company. Market analysts, consultants, and in-house market research departments try to track their industry's activities using traditional methods. Corporate research typically tracks newspaper articles, online databases, annual company reports, and competitors' Web pages and then presents the results to management.

There are several problems with the conventional research process:

- Information archives are document-based. For a collective gathering of facts, this is a document-centric view that is too coarse to be useful.

- Typical document management systems rely almost exclusively on information retrieval techniques that are inaccurate.
- Implications can only be made transparent if you use background knowledge, but systems today rarely support background knowledge.
- Different people might contribute knowledge.
- Different people might require different views of the same basic piece of information.

A KM system that covers such knowledge about the outside world should

- support the collective gathering of information on the level of facts rather than documents,
- integrate the gathering task smoothly into the common research process,
- allow you to intelligently combine facts,
- check new facts against the available background knowledge,
- allow multiple-view access to the knowledge through a single entry portal, and
- allow you to route derived facts back into the common workplace environment.

With these aims in mind, we developed an ontology-based application called the Corporate History Analyzer (CHAR).

Kickoff phase

The question was how to bring the required conceptual structures and reasoning capabilities into action following the further steps of our knowledge metaprocess shown in Figure 6. We collected user requirements in the requirements specification phase during structured interviews with corporate research analysts. The Ontology Requirements Specifications Document shown in Figure 7 is the product of this first phase.

We specified the ontology and then developed it in detail. We asked the domain experts what questions they would expect to be answered by a system supporting their corporate research work. These competency questions helped us find the most important concepts and relations between them.

Analyzing the questionnaires helped us conclude that the system should deliver answers about the acquisitions, mergers, and restructuring of companies over specific time periods. The questionnaire in Table 2 lists competency questions CQ1 to CQ4 that we compiled during the ontology kickoff. We also recognized the need for a clarification of organizational wording, so we could more clearly determine, for example, whether a business unit is a division or a department.

Refinement phase

In the refinement phase, we brought the concepts into a taxonomic hierarchy designed for the knowledge engineer. We showed the taxonomy—a part of which is shown in Figure 8—to domain experts to generate additional concepts and relevant

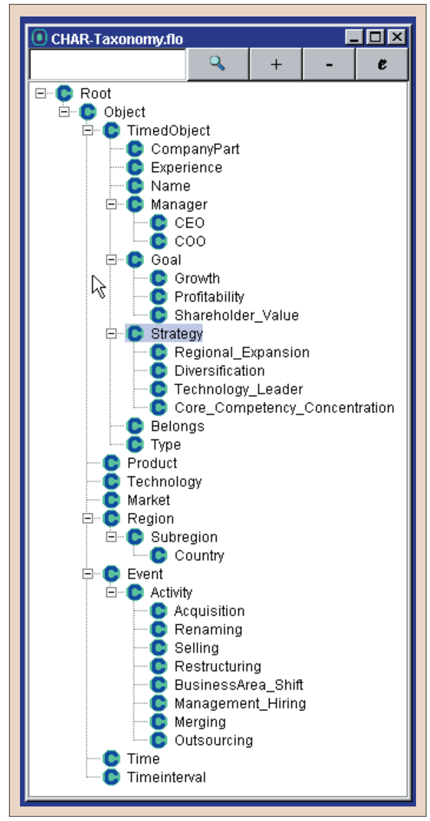


Figure 8. The Corporate History Analyzer taxonomy.

Table 2. Sample competency questionnaire for a business strategy in the chemical industry.

Domain: Business strategy in the chemical industry
Date: 2000/11/26
Ontology Engineer: T. Model

CQ no.	Competency questions	Concepts	Relation
CQ1	What are the subsidiaries, divisions and locations of company X?	company, subsidiary, division, location	company *has* subsidiary company *has* division company *has* location
CQ2	Which companies acquired company X?	company, acquisition	company *makes* acquisition acquisition *has* buyer acquisition *has* seller
CQ3	Which companies merged in 1990 in the rubber industry?	company, merger, year, industry	company *makes* merger company *isPartOf* industry merger *happensIn* year
CQ4	Who is CEO of company X?	CEO, company	company *has* CEO
CQ5	Which activity of company X leads to operation in region Y?	activity, company, operation, region	company *performs* activity activity *leadsTo* operation operation *takesPlaceIn* region
CQ6	Is there any regional expansion of company X due to the acquisition of company Y?	expansion, company, region, acquisition	company *makes* expansion company *makes* acquisition expansion *takesPlaceIn* region
CQ7	...		

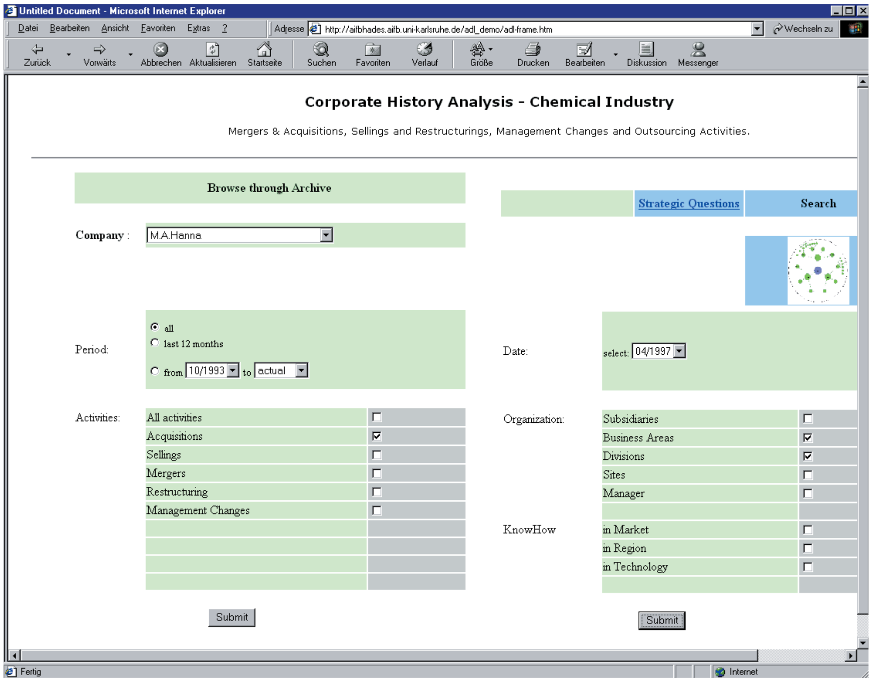


Figure 9. The Corporate History Analyzer.

attributes, and to establish relationships between the concepts. OntoEdit, our ontology engineering tool, supported this work.[10] OntoEdit allows you to model the ontology and formalize it in different representation languages, like F-Logic and DAML+OIL.

We found that in the corporate history domain, time had to be modeled. A company has a beginning time and (potentially) an end time, which would be when the company gets acquired by another company, merges with another company, or goes bankrupt. So we found we had to model duration.

One of the key characteristics of CHAR is that the user provides facts to the system about actions like acquisitions and mergers and the system then infers the consequences. To do this, you have to model rules for all possible activities that incur such consequences. For instance, a rule might read like this: "If two companies are merged, a new company with a new name is created, the old companies cease to exist, they are from now on subsidiaries of the new company" or "If a division is outsourced it becomes part of a new company or it becomes a company on its own."

For CHAR, we had to develop a Web interface that used the ontology for querying and augmenting the knowledge repository (see Figure 9). We had to perform a query development step to formalize the views and competency questions described earlier. This development step depended mostly on the actual application setting. It was independent from the ontology construction.

Evaluation phase

In the evaluation and testing phase, we investigated the ontology's usability. With regard to the CHAR ontology, we found that the competency questions CQ1 to CQ4 (Table 2) could be handled successfully by the ontology. However, we wanted complex knowledge content to be queried and depicted on one screen. This kind of content would include company purchases, regions of previous activities, and regions of activities of the purchased company.

The research analyst found that singleton knowledge items were spread over different screens of the application. They were very hard to combine by the research analyst to answer strategic questions. He wanted better support from the ontology. Through the discussion, we came up with additional competency questions (CQ5 and following) that should be handled by the ontology. We then added corresponding axioms to the ontology and introduced a new query window for strategic questions.

Maintenance phase

We are currently in a maintenance phase, where through a shift in the goal orientation of the application, we face requirements for considerable extension of the ontology and the application. In addition to corporate histories, the system must now support the extended tracking of market circumstances, including a more detailed view of available technology and better comparisons with peer groups.

Major problems that we now must face include the documentation and versioning problems that have so far been neglected by practically all ontology development tools. We are working toward resolving these issues in our ontology environment OntoEdit.

We developed our methodology parallel to CHAR. Thus, we could not follow all the methodological steps we've presented here from the very beginning. Developers who are new to the project need to reverse-engineer some old ontology parts to find answers about how these parts actually work. We expect that comprehensive use of the proposed methodology and its full-fledged support by OntoEdit will mostly prevent these problems in the future.

Knowledge process instantiated

CHAR should allow many people to contribute factual knowledge in a way that can be embedded into their common work process and organized around a semantic background. It should also provide multiple views onto the same knowledge base for different time frames, for different regional foci, for varying intra-organizational structures, and for different strategic questions, to name but a few.

Providing knowledge

Providing new facts for the knowledge repository should be as easy and as smoothly integrated into common work as possible. There are several ways to do this:

- You can enter information through a form-based interface.
- When information is produced during documentation or report-writing, you can use a template approach to generate knowledge. Both of these tools affect the knowledge creation step.
- You can use wrapper mechanisms to provide data from tables and lists on the Web (import step).
- Most important for CHAR, you can use our annotation tool to add metadata to data given in documents, which affects the capturing step.

Figure 5 shows a snapshot of OntoAnnotate.[4] The user works with documents using a text or spreadsheet tool or an Internet browser. When the user detects some relevant change being described in the document, he or she highlights the word or phrase and uses the annotation tool to select the type of the phrase (like "M.A.Hanna is a company") and its relation to other material (like "Hanna sells Shapes Distribution Business to GE Plastics on 11 May 2000"). The knowledge repository stores the document, these facts, the time of annotation, and metadata about the annotator.

Querying for knowledge

We designed CHAR's query interface to deal with organizational and strategic questions that depend on spatiotemporal constraints. CHAR renders views that can be seen on a common Web browser. Actually, these views look just like common Web sites. The available menu selections are controlled by the knowledge of the back-end system. For example, you can select companies known to exist in the knowledge repository. Figure 9 shows the main views offered by CHAR. These views include Activities, Organization, Know-How, Strategic Questions, and General Query Possibilities (indicated by "Search").

The first major category of queries relevant to the corporate history is about organizational structures and the activities that change organizational structures. For instance, the view of "Acquisitions of M.A.Hanna" returns all its purchases. CHAR offers corresponding views for Sales, Mergers, Restructuring, and Management Changes (see Figure 10).

What is interesting to note at this point is that it is rather difficult to get a clear picture of what is really happening with M.A.Hanna. It is difficult and time-consuming for the human analyst to detect some trend in lists of single knowledge items. Observations become much easier when different types of knowledge items can be related and contrasted.

For instance, Figure 10 shows two snapshots of the organization of M.A.Hanna. They are automatically derived from single activities, like acquisitions and restructurings, and they can give the analyst a neat picture of how formerly isolated purchases that M.A.Hanna made before 1994 were more tightly integrated in the company in 1997 (for example, "Southwest Chemical" having been reorganized into the Business Area "Plastic Compounding").

In addition to sophisticated support based on concrete facts and figures, CHAR supports strategic questions that indicate possi-

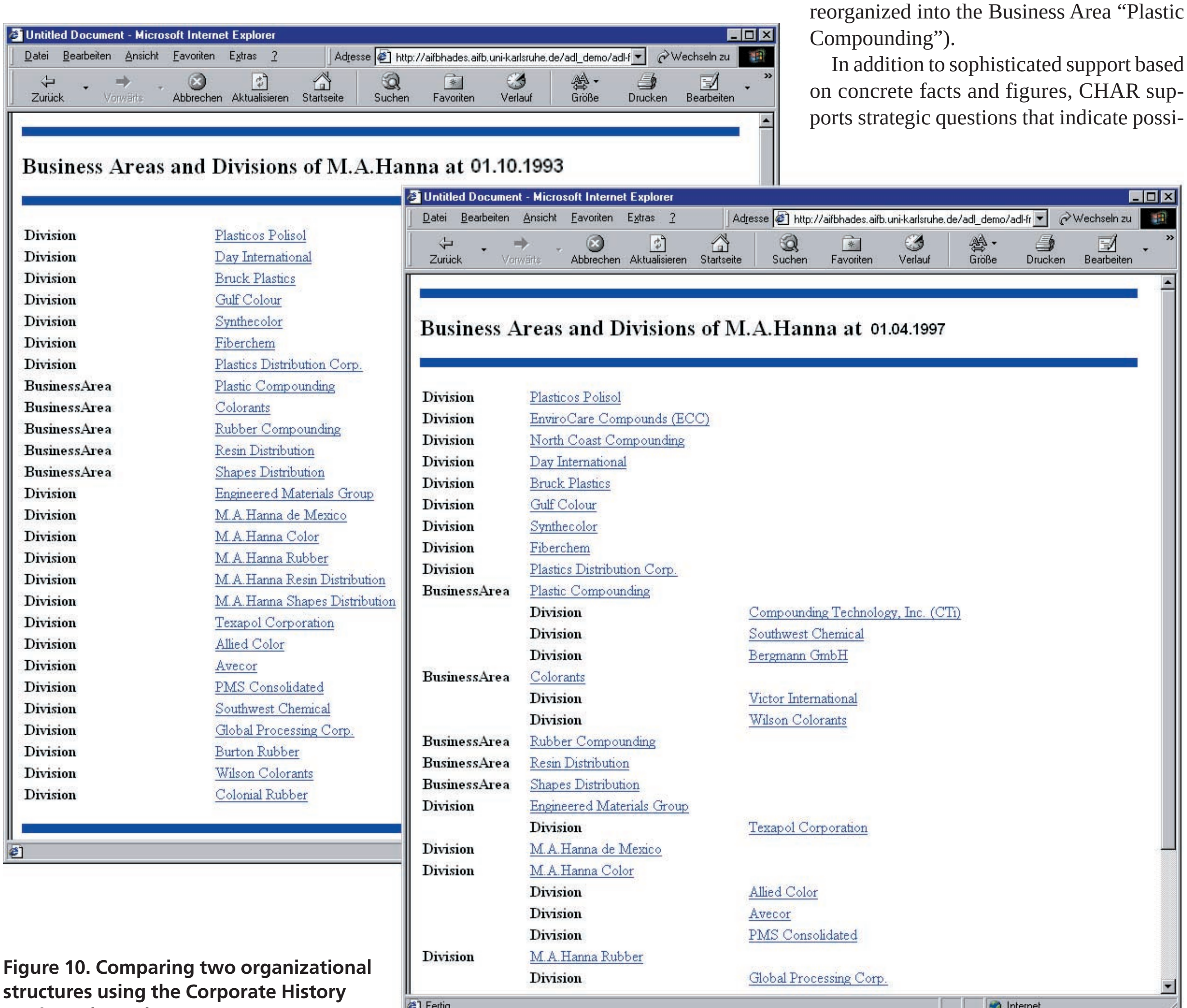


Figure 10. Comparing two organizational structures using the Corporate History Analyzer (CHAR).

ble answers to questions about business competitors that cannot be answered definitely and often rely on conjecture. For instance, the purchase of a company from abroad might lead to a gain of market share in that area, and thus to a regional diversification.

In the future, more and more companies will find that analysis of knowledge processes will feed back into the knowledge metaprocess cycle and can help to improve both. These companies will find that new concepts that come up during the use of a KM solution can be introduced back into the ontology to help it evolve. For this purpose, we investigate the use of semi-automatic ontology learning techniques.[10]

The challenges lie in analyzing KM processes and dynamically updating the KM solution. Our framework can help leverage these evolving systems by providing a concise view of the problem.

Related to this work, we are jointly organizing the first German conference on knowledge management, the WM2001 (http://wm2001.aifb.uni-karlsruhe.de). Steffen Staab and Rudi Studer are the cochairs, York Sure is the industrial liaison, and Hans-Peter Schnurr is the finance chair.

Acknowledgments

We thank our colleagues and students at AIFB, University of Karlsruhe, VU Amsterdam, and Ontoprise GmbH, including H. Akkermans, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, M. Maier-Collin, D. Oberle, and D. Wenke, without whom the research would not have been possible. The research presented in this paper was partially funded by EU in the IST-1999-10132 project On-To-Knowledge and by BMBF in the project GETESS (01IN901C0).

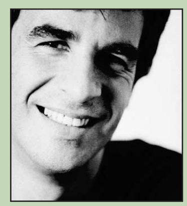
References

1. R. Dieng et al., "Methods and Tools for Corporate Knowledge Management," *Int'l J. Human–Computer Studies*, vol. 51, no. 3, 1999, pp. 567–598.
2. M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications," *The Knowledge Eng. Rev.*, vol. 11, no. 2, 1996, pp. 93–136.
3. G. Schreiber et al., *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, Cambridge, Mass., 1999.
4. S. Staab et al., "Semantic Community Web Portals," *Proc. WWW-9/Computer Networks*, Elsevier, New York, vol. 33, 2000, pp. 473–491.
5. L. Morgenstern, "Inheritance Comes of Age: Applying Nonmonotonic Techniques to Problems in Industry," *Artificial Intelligence*, no. 103, 1998, pp. 1–34.
6. A. Abecker et al., "Toward a Technology for Organizational Memories," *IEEE Intelligent Systems*, vol. 13, no. 3, May/June 1998, pp. 40–48.
7. S. Staab and D. O'Leary, eds., *Bringing Knowledge to Business Processes*, 2000 AAAI Spring Symposium technical report SS-00-03, AAAI Press, Menlo Park, Calif., 2000.
8. D. O'Leary, "Knowledge Management: An Analysis Based on Knowledge Use and Reuse," *IEEE Intelligent Systems*, vol. 16, no. 1, Jan./Feb., 2001.
9. M. F. López et al., "Building a Chemical Ontology Using Methontology and the Ontology Design Environment," *IEEE Intelligent Systems*, vol. 14, no. 1, Jan./Feb. 1999, pp. 37–46.
10. A. Maedche and S. Staab, "Mining Ontologies from Text," *Proc. 12th Int'l Conf. Knowledge Eng. and Knowledge Management*, Lecture Notes in Computer Science, vol. 1937, 2000, Springer Verlag, New York, pp. 189–202.

The Authors

Steffen Staab is an assistant professor at the University of Karlsruhe and cofounder of Ontoprise GmbH. He received an MSE in computer science from the University of Pennsylvania and a Dr. rer. nat. from the University of Freiburg, Computational Linguistics Lab. He has chaired several international workshops, including the AAAI 2000 spring symposium on Business Processes and KM. Contact him at Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany; sst@aifb.uni-karlsruhe.de.

Rudi Studer is a professor in applied computer science at the University of Karlsruhe. His research interests include knowledge management, intelligent Web applications, knowledge engineering, and knowledge discovery. He received a doctorate in mathematics and computer science and a Habilitation in computer science at the University of Stuttgart. He is a member of IEEE, ACM, AAAI, and GI. Contact him at University of Karlsruhe, Institute AIFB, D-76128 Karlsruhe, Germany; studer@aifb.uni-karlsruhe.de.



Hans-Peter Schnurr is managing director of Ontoprise GmbH, a company providing a wide range of semantic Web technologies. His current research interests include semantic Web and ontology-based applications. He received a diploma in industrial engineering and business economics from the University of Karlsruhe. He is cofounder of the German association for knowledge management (GfWM) and member of the German Association for Computer Science (GI). Contact him at Ontoprise GmbH, Haid-und-Neu Str. 7, 76131 Karlsruhe, Germany; +49(0) 721 6635933; Fax +49 (0) 721 6635934; schnurr@ontoprise.de.

York Sure is a PhD candidate at the Institute of Applied Computer Science and Formal Description Methods at the University of Karlsruhe. His current research interests include knowledge management, semantic Web, and ontology-based applications. He received a diploma in industrial engineering from the University of Karlsruhe. He is a member of the German association for knowledge management (GfWM) and the German Association for Computer Science (GI). Contact him at Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany; sure@aifb.uni-karlsruhe.de.