A Business Model in a Web Representation Used as an Organizational Memory

Hermann Kaindl
Siemens AG Austria, PSE
Geusaugasse 17
A-1030 Vienna
Austria
Phone: +43 51707 43288
Fax: +43 51707 53270
Email: hermann.kaindl@ieee.org

In the context of knowledge-intensive work, fluctuation of employees may involve the loss of a key asset. New employees often have to painfully learn from experienced ones when the key knowledge is *tacit*. In the spirit of *corporate knowledge management*, we present a bottom-up approach to modeling business knowledge for an *organizational memory*. The implemented Web representation in the intranet of our company based upon this approach provides both artifacts like documents and business process descriptions (that people are supposed to follow daily). We currently apply the same idea also in two external organizations.

Keywords: business process, organizational memory, tacit knowledge, object-oriented modeling, hypertext, Web representation.

Introduction

When work complexity is high in the given work setting, people need much knowledge to get their job done [Davenport, 2000]. In such business areas, experienced employees are the true asset of the organization. Therefore, fluctuation of employees raises serious problems:

- When experienced people leave the organization, important knowledge about their work processes may be lost.
- Making new employees familiar with the daily work processes (especially more knowledge-intensive ones) is costly and takes its time.

While there is no silver bullet for solving this issue, *corporate knowledge management* (CKM) addresses it, among other things, by maintaining an explicit *organizational memory* (OM). Ideally, the OM would store the relevant knowledge and make it accessible to the people needing it for their work.

For the creation of such an OM, *business (process) models* of such work processes are needed. Unfortunately, the usual approaches to business modeling focus on a top-down understanding through managers and/or analysts only. The resulting high-level models are typically used by the management for changing the business rather than by the people for running it.

In this paper, we propose to create and use a business process model bottom-up as an organizational memory to be used by the employees, i.e., the people supposed to work according to this model (including management, but not exclusively). We focus on the knowledge management issue of making this *tacit knowledge* [Polanyi, 1966] of the business explicit (through *externalization* [Nonaka and Takeuchi, 1995]) and accessible to the people. When people have an OM with an explicit and easily accessible representation of the business process model available, according to which they are supposed to work, they may have a better basis to do so and more ownership of these business processes and also their changes.

We actually built such a business model. The goals of this work were to have

- a complete and consistent description of the processes in the given business area;
- a definition and description of the “business functions”;
- a representation of the various links among those entities; and
- accessibility by all the people working in this business area through the intranet.

For building this application, the knowledge-acquisition approach to convert tacit knowledge to explicit knowledge was object-oriented modeling. In contrast to diagrams like UML (Unified Modeling Language) [Rumbaugh *et al.*, 1999][1], that may not be well understood by all kinds of employees, we supported it with a hypertext/hypermedia tool, where the hypertext structures are organized in classes, subclasses and instances. Any kind of files, including documents and sound, for instance, can be attached to such hypertext nodes. Since several people have been working on this knowledge acquisition task concurrently, we pre-defined a template according to an object-oriented metamodel. The resulting hypertext is fully automatically exported to a Web representation (in HTML and using ASP scripts), where it is accessible in the company's intranet. This Web representation provides easy and current *access* for the people to the OM (supporting *internalization*), which helps to improve the efficiency of their work.

This paper is organized in the following manner, with a focus on building such an organizational memory. First, we sketch our approach to knowledge modeling. Then we elaborate on the use of this approach for the externalization of tacit business knowledge in order to build the organizational memory. We also illustrate its form as given in our company's intranet. Finally, we summarize a few lessons learned and provide an outlook on the application of the same idea in two external organizations.

Our Approach to Knowledge Modeling

Basically, our approach to model the business knowledge in this project was object-oriented modeling. Usually, object-oriented models are represented in a graphical notation, the standardized and most widely used one now being UML. Still, however, many people outside of object-oriented software engineering are not really familiar with UML diagrams. So, representing the OM in UML would have involved the risk that many people simply were not able to understand its language and, therefore, also not its content.

Instead, we used or own approach RETH (Requirements Engineering Through Hypertext), which was originally developed for acquiring and representing requirements. So, we need to review some of the features of this tool and its supported method. RETH's basic approach (and using hypertext for it) is described in [Kaindl, 1997], and the tool's mechanism for semi-automatic link generation in [Kaindl *et al.*, 1999]. Scenarios, goals, domain entities, etc. are

[1] The standardized specification of UML is available at the time of this writing at http://www.omg.org.

all modeled as objects in RETH. Classes and instances are distinguished, and they are described in one hypertext node each in the tool. In this way, descriptions in natural language of, e.g., requirements statements or glossary entries are integrated in the object-oriented structure. Classes can have superclasses and subclasses, which is useful for modeling generalization/specialization hierarchies (taxonomies). In fact, the RETH tool can make itself concrete suggestions for creating such a hierarchy. The tool also implements *inheritance* in such hierarchies. The hypertext nodes in RETH have internal structure through so-called *partition*s, which can represent object attributes, associations and aggregations. Based upon hyperlinks, the tool can make itself suggestions for associations [Kaindl, 1996].

Since this approach covers both specialization and aggregation, it can be used to model a domain like business processes along both these dimensions as illustrated by the "Process Compass" in [Malone *et al.*, 1999]. Therefore, it also supports the navigation through richly interconnected networks of related processes and activities using such a compass.

Fig. 1 illustrates a typical window of RETH's user interface. Its left part shows a hierarchy of hypertext nodes according to the class/subclass and instance structure. Each node is either labeled with the symbol "C" for class or the symbol "I" for instance. The right part of the window shows the content of the node that is selected in the left part. Each partition of the node can be expanded (through selecting "+") or shrunk (through selecting "–"), depending on its current state in the user interface. As an example, Scenario description is currently shown expanded. It contains a textual description of a typical sequence of actions for requesting hardware, which includes hypertext links, e.g., to Project leader. Instead of text, a partition may contain attachments of any kind of file (including pictures, sound or video), which makes the hypertext actually a simple form of *hypermedia*.

RETH runs on standard PCs under Windows 95, 98, NT and 2000. The status of the tool is like that of a product (beyond alpha and beta versions), but it is not sold as such. Instead, it is used internally and in the course of consulting (also externally).

Additionally, RETH denotes a *method* supported by this tool. A process of scenario-based requirements engineering can be found in [Kaindl, 2000], which builds upon the *metamodel* of this method. It defines which objects are to be represented and which relations between objects. For example, goals and scenarios are to be represented as well as a many-to-many relationship between them. It associates in the model goals with those scenarios whose execution achieves them. As an example, Fig. 1 shows the object-oriented *link* between the scenario instance Requesting hardware and the connected goal instance Hardware available according to the association Achieve (as represented in the RETH user interface). This means that Hardware available can (normally) be achieved by Requesting hardware. It can (normally) also be achieved by Exchanging hardware, as an alternative means.

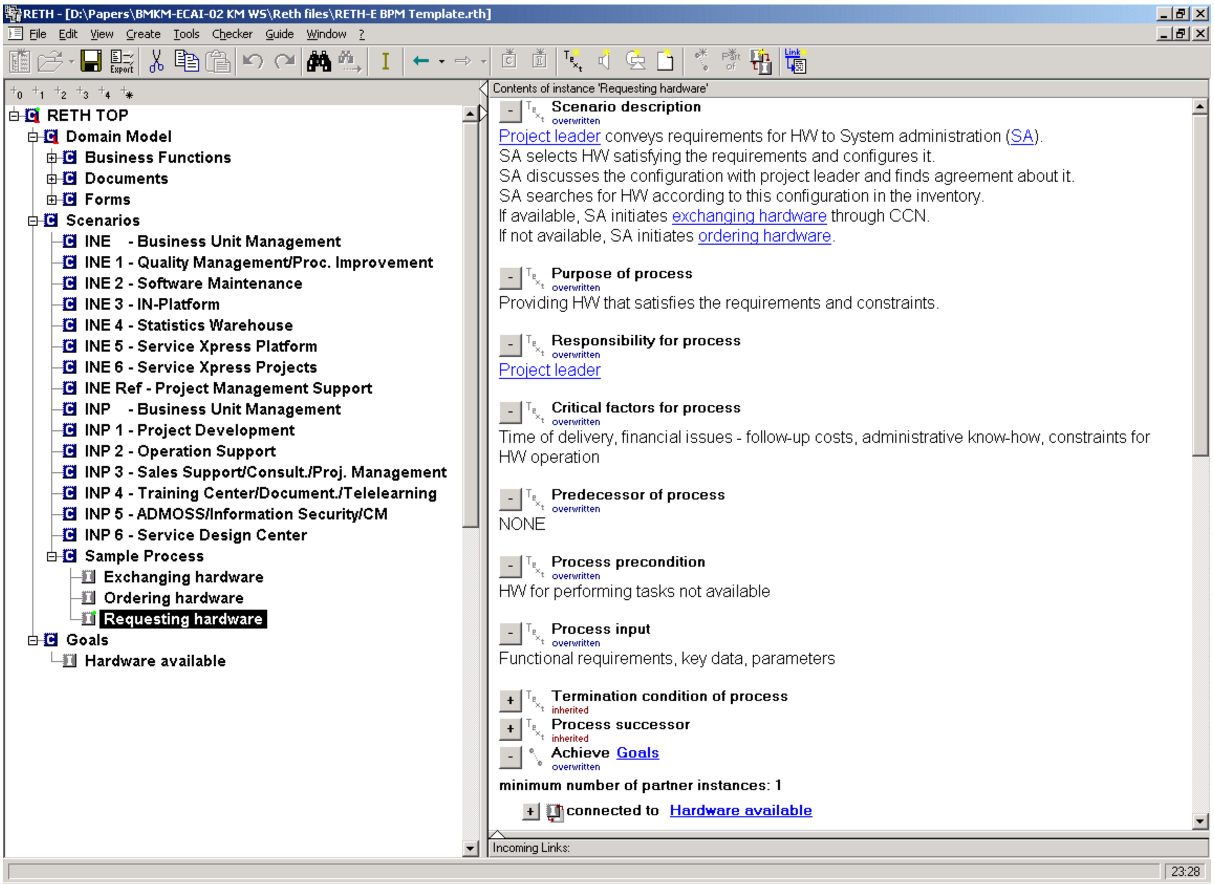

Figure 1. A screen dump of a typical RETH window illustrating the template.

In order to help people getting started in using the tool in a purposeful way (according to the method), we devised and implemented a *guide* [Kaindl *et al.*, 2001]. Its support covers both the conceptual level to represent scenarios, goals and their relationship and the more technical level of how to do that with the given tool.

In the object-oriented community, scenarios are typically dealt with in the form of *use cases*, which can be viewed as classes of scenarios. In fact, use cases have been proposed for business modeling previously [Jacobson *et al.*, 1994]. Our approach to business modeling is analogous from a conceptual point of view, although it is transferred from requirements engineering. As argued above, however, is does not center around object-oriented diagrams and, it has more focus on goals.

According to our best knowledge, using such an approach for building an OM for CKM is new. In the given project, the rationale for the decision in favor of our approach was the following:

- object-oriented modeling, which the people involved were familiar with;
- machine-support for hyperlink generation [Kaindl *et al.*, 1999], which made creating links very inexpensive;
- possibility of separate development through several people concurrently, which was necessary due to the involvement of different departments;
- automatic export facility for the intranet, which was the distribution medium of choice; and
- in-house consulting (through the author of this paper), which was easily and quickly available and less expensive than external consulting.

So, object-oriented modeling was deemed to be useful, but several of these arguments relate to our tool and its underlying hypertext approach as well.

Creating the Organizational Memory

Based on this approach to knowledge modeling, an OM containing a business model has been created in our environment, a large (software and hardware) systems development division of a large multi-national company. The departments involved work in the field of telecommunications. Since nearly fifteen departments have been involved, this task had to be split into pieces, one for each department and one for each of the two business units containing these departments. Creating the OM was initiated in the course of a so-called business improvement project. The main incentive for the people to model their processes was increased reputation and praise for contributing to this prestigious project.

Core structure as a template

Before the real work of building this business model was started, a small team including the author of this paper built a core structure as a template for two reasons:

1. providing help for getting started;
2. facilitating uniformity of representation of the various business processes.

Fig. 2 illustrates part of this core process structure through a UML class diagram. (This diagram was automatically generated from the RETH representation illustrated in Fig. 1.) While the real process structure was defined in German, we explain it in English in order to facilitate its understandability in the context of this paper. This diagram shows that Scenarios and Goals are related to each other as sketched above for the RETH metamodel (and illustrated in Fig. 1). This relation is represented here through an object-oriented association.

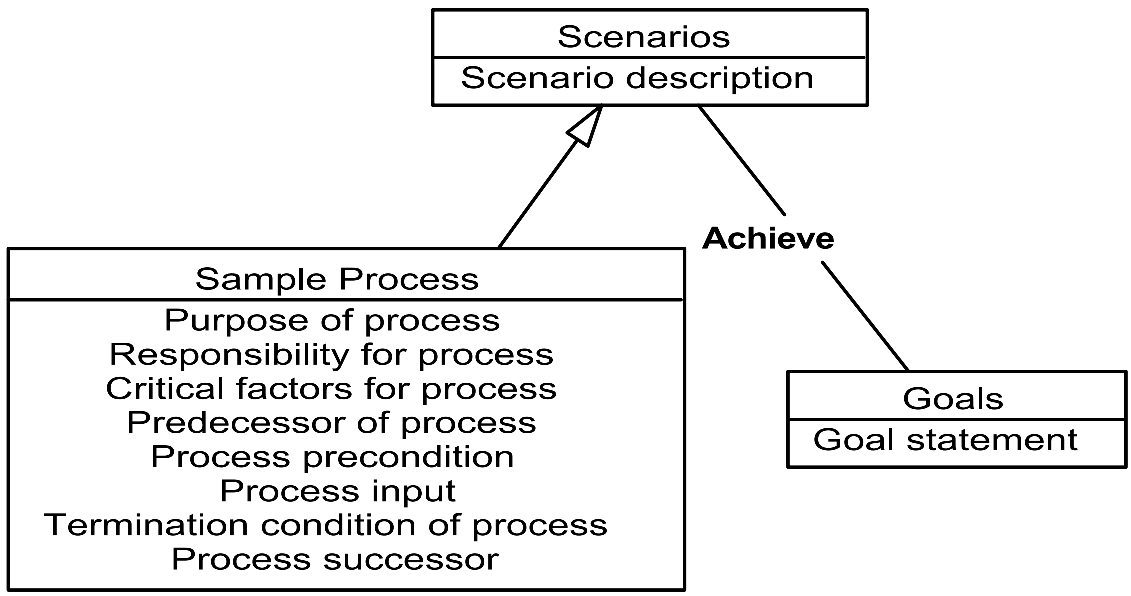

Figure 2. A UML class diagram of part of the core process structure.

Since concrete examples are usually easier to understand than theoretical explanations, this template includes a sample process as well. Fig. 2 shows that the class Sample Process is a specialization of the class Scenarios. That is, the sample process is a special class of scenarios. In the RETH tool, this is shown through indentation (see Fig. 1 again).

Based on this specialization relationship, Sample Process *inherits* the object-oriented *attribute* Scenario description from Scenarios. In addition, it has its own attributes like Process precondition for the characterization of the process. The RETH tool can show both inherited and newly defined attributes (on the right side of its window). Sample Process also inherits the association with Goals, which means that the relations between processes and their goals can be explicitly represented. Requesting hardware is an instance of the sample process and therefore (indirectly) of the class of scenarios (see Fig. 1 again).

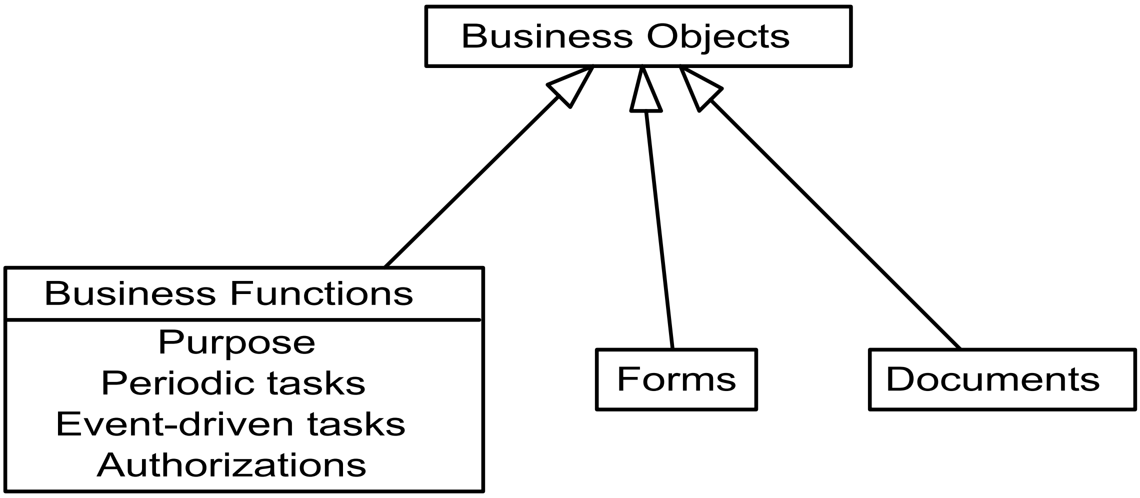

Figure 3. Classification of business objects.

Fig. 3 illustrates another part of the core structure, a classification of business objects. Business Functions (e.g., business unit manager or project leader), Forms (including document templates and check lists) and Documents are specializations of Business Objects. Only for Business Functions, the core structure predefines general attributes for their characterization:

- Purpose (in the business),
- Periodic tasks,
- Event-driven tasks, and
- Authorizations.

While the use of UML notation primarily served the purpose of specification, such diagrams were given to the developers of the OM as well. Still, their representation to work with was the object-oriented hypertext in the RETH tool (see Fig. 1 again, where Domain Model is given instead of Business Objects, as explained below). This was also the basis for the Web representation of the OM to be finally used.

After the template was developed, it was reviewed and internally released.

Knowledge externalization

While a few informal process descriptions existed already, the major part of the business model had to be made explicit from tacit knowledge about the daily work, which is called *externalization*. For this task, the template described above was provided for all people supposed to model the process in their part of the business area: department managers or employees which they authorized to do the job on their behalf. Due to the predefined structure and the exemplary text from the sample process, they did not have to figure out on their own in which form the processes were to be described.

The uniform and clear structure facilitated a uniform appearance of the overall business model. In addition, we organized tutorials for introducing those people to the approach, the template and the RETH tool even before they started their work.

After that, several people have been working concurrently using the tool, each one on his or her part. This was facilitated by the divide-and-conquer approach of partitioning the OM into parts for each organizational unit involved.

They also brought in additional information in the form of traditional documents (in MS Word or simple ASCII text), emails, spread sheets and MS PowerPoint presentations. When a traditional document already captured some of the process knowledge, it was attached to the node where it was newly structured, supporting *traceability*. Other information that is better described in various kinds of files comprised the following:

- concrete forms and checklists,
- information about tenders,
- documentation about regulations,
- reports, etc.

After some parts of the overall process model have been more or less completed, we began to install a *glossary index* to the most important pieces of information. Since the content of the OM is subject to changes, enhancements and updates, working on this index is also an ongoing activity.

Whenever significant changes have been made to the content of (part of) the OM in the RETH tool, the updated OM is made accessible in the intranet. Such an update simply involves connecting the separate parts, exporting them fully automatically to a Web representation and uploading it to the Web server.

The Resulting Organizational Memory Accessible in the Intranet

In principle, it would have been possible to install the resulting OM in the form of a RETH hypertext/hypermedia repository on each employee's desktop. This would have meant, however, to make them all familiar with this tool, at least for browsing. This should not have been too hard, because RETH's user interface is similar enough to well-known Web browsers and MS Office products. But having to use yet another tool is always an obstacle to adapting to the new and unfamiliar. In addition, everyone involved would either have to get access to the same file server, or other strategies for distributing new versions of the OM would have been needed.

So, the more appropriate and timely fashion to provide access to such an OM seems to be making it available in the intranet. Since providing a machinery for direct access to RETH through the intranet would have been more expensive and was not even necessary, we employed the already existing export facility of RETH for HTML pages. In fact, this facility provides a look-and-feel similar to that of the RETH tool through the usual Web browsers (for browsing of a snap-shot only). The hypertext links in the RETH tool are just normal URLs in HTML. The explorer-style representation including the hierarchy is provided through an implementation via *active server pages* (ASP). Much as in the RETH tool itself, each file attachment is linked into the HTML representation. While most kinds of pictures are directly embedded in place, for all other kinds of attachments dedicated programs are called (based on their file extensions).

In order to give an idea of the size of this OM, it currently contains about 400 hypertext nodes (one per HTML file) and more than 60 attached files. For illustrating it, we show several screen dumps from its appearance in MS Internet Explorer. While the text is in German, especially the structure should be clear from the explanations above.

Fig. 4 illustrates the representation of the business function GF-Leiter (business unit manager) in the OM, including its glossary entry (directly below the title) and its attributes. These are the attributes (as named in German) which are defined in the core structure for Business Functions in general (see Fig. 3 above), but filled in with specific text.

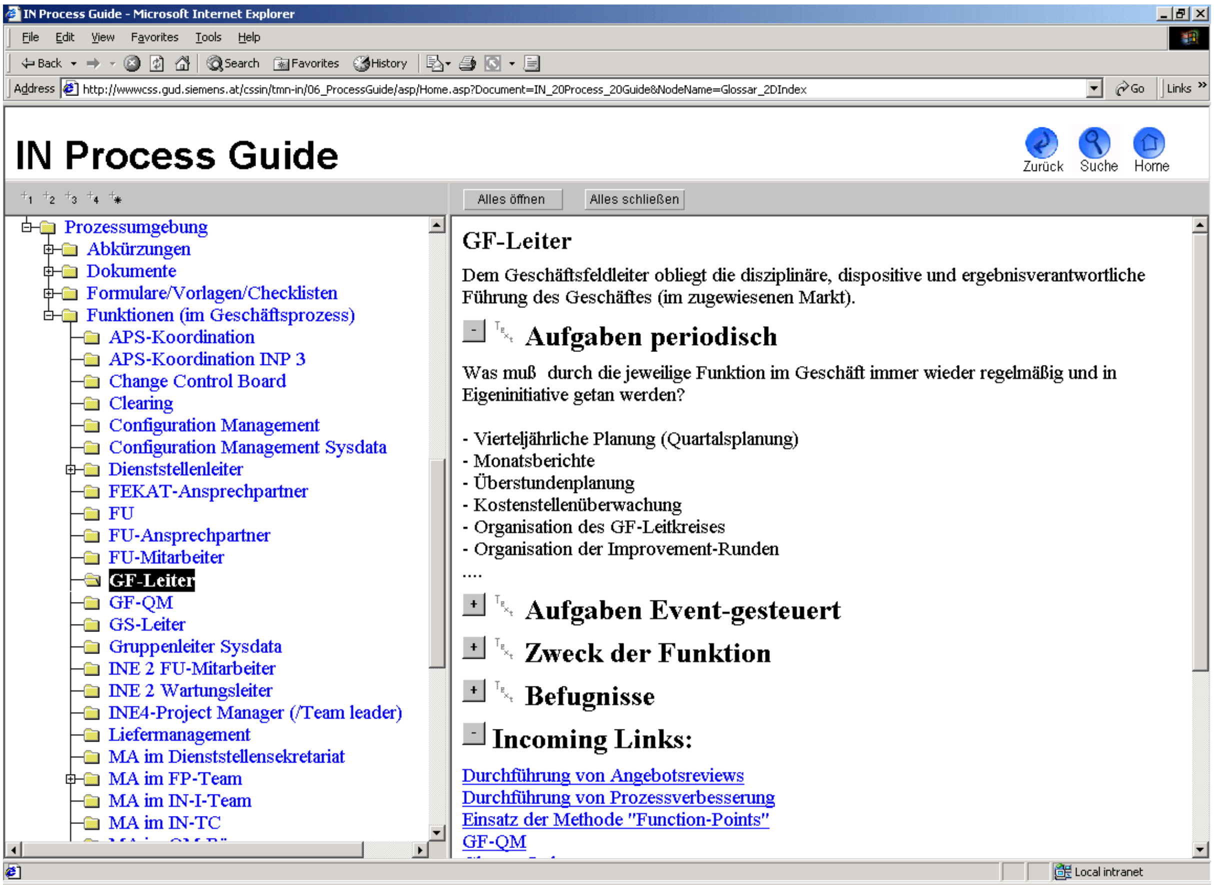

Figure 4. A screen dump of a Web-browser window illustrating a business function.

Much effort has been spent by the developers of this Web representation to make it look much like the user interface of the desktop application (compare Figs. 1 and 4 in this respect). In fact, also the Web representation had the same icons for classes and instances as the desktop application. End users of the Web representation, however, turned out to be less interested in the fact that this indicates an object-oriented model and often did not even understand the symbols of these icons. So, we made this representation configurable at the time of the export and use mostly the configuration with folder icons now as shown in Fig. 4.

While the hypertext links in the RETH tool are bidirectional (i.e., they can be traversed in the opposite direction as well), URLs are inherently unidirectional. In order to provide the same navigational options in this regard also in the Web representation, RETH automatically generates extra URLs for the other direction as well. Fig. 4 shows some of them under Incoming Links.

Process descriptions look much the same as business functions in this representation. While the latter are structured according to Fig. 3, however, process descriptions are structured according to the Sample Process in Fig. 2.

Fig. 5 illustrates an example of a document entry (Wochenbericht, which is a weekly report). It has an MS Excel file attached, which is simply opened by clicking on the file name. Due to its file extension .xls, this application is called for opening the attached file, which MS Internet Explorer does in an extra window.

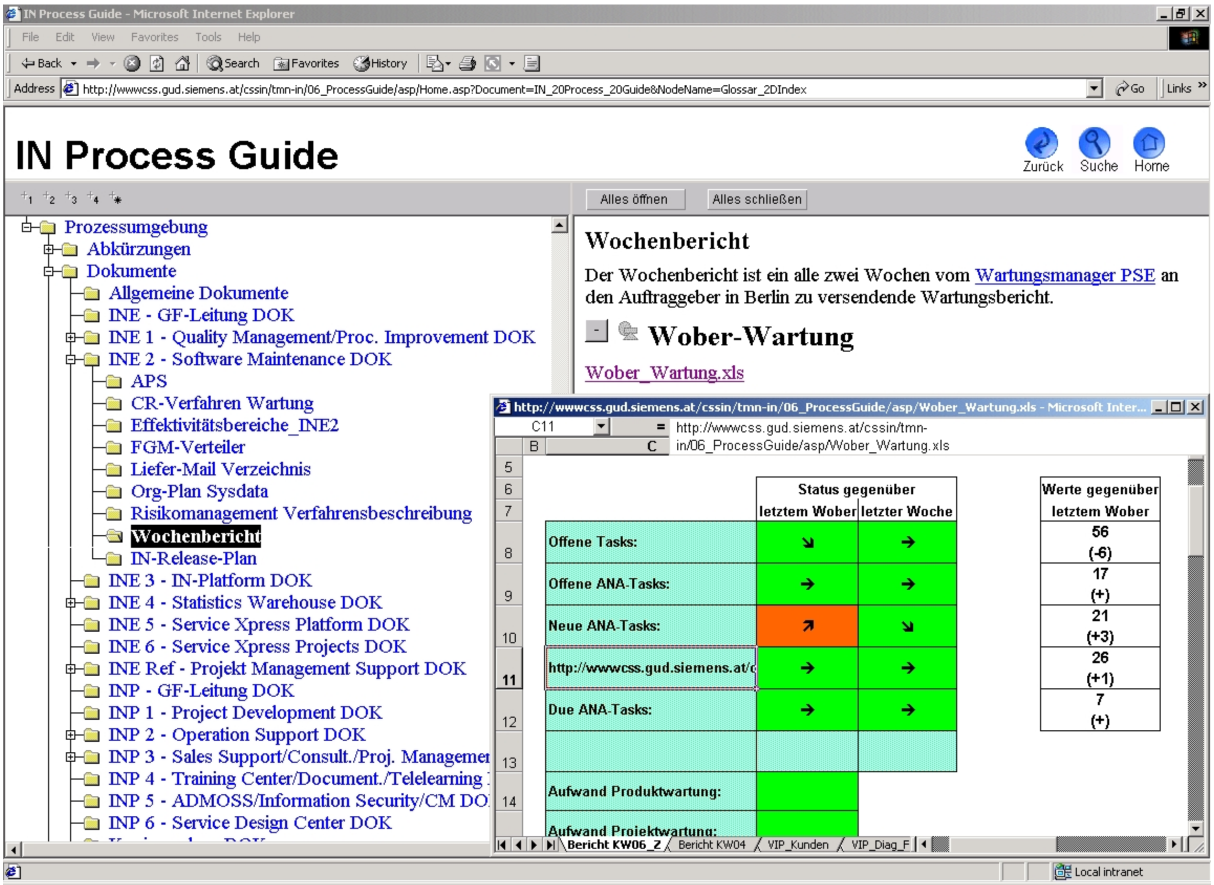

Figure 5. A screen dump of a Web-browser window illustrating a document entry with a file attachment.

Fig. 6 illustrates an example of a file attachment containing a picture (in .jpg or .gif). Such pictures are directly embedded in place (in contrast to opening a new window as illustrated in Fig. 5 for other file attachments). While we have argued against using a graphical language like UML for exhaustive representation of business processes in such an application, diagrams and pictures can, of course, help to get a quicker and better understanding of certain ideas when added to textual descriptions. Whether or not such a diagram is useful for the people will largely depend on their familiarity with the notation.

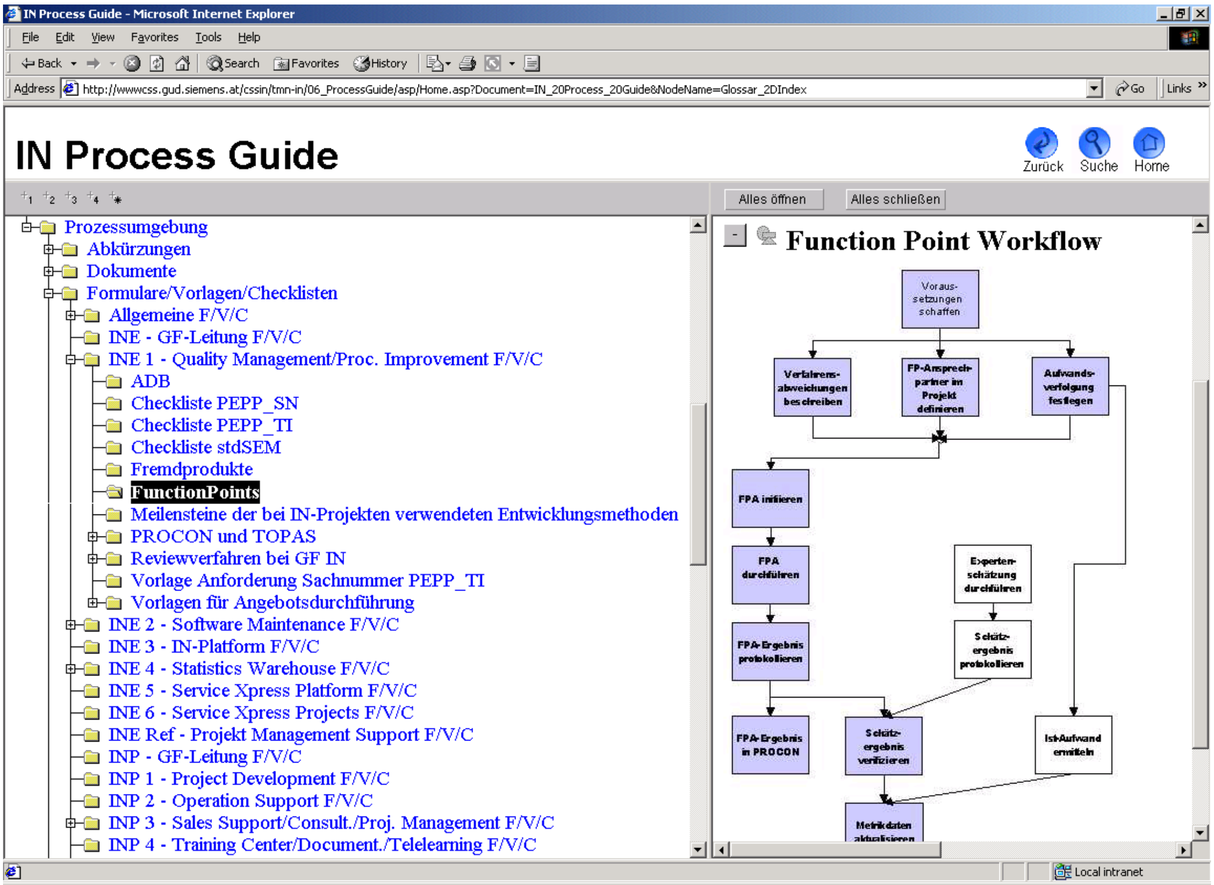

Figure 6. A screen dump of a Web-browser window illustrating an embedded picture.

A major issue with such a representation of an OM in the intranet is finding the relevant pieces. Therefore, we offer three possibilities of *access*:

1. glossary index,
2. string search and
3. specialization hierarchy.

While the first two of them are fairly standard, the access through the specialization hierarchy seems to be unique in our approach.

Explanations of the purpose of this OM, its structure, user interface and the navigational possibilities are given in extra Web pages. These are linked to from the same page from where the OM itself is accessible.

Lessons Learned

According to observations, this OM has been used primarily for guidance on both rather trivial tasks (like registering for tutorials and seminars) and more intricate and important processes (like preparing and reviewing tenders). Unfortunately, we do not have experimental data about the usefulness of our approach. Still, let us summarize a few generalized lessons learned from observations and feedback:

- *Providing a template is useful.*

  First, it was not too difficult to include the already existing process descriptions for parts of the overall business process covered here. The files of the traditional documents containing them were attached, and their content elaborated and structured according to our template. This template provided both an appropriate form and helped to find missing spots in the given descriptions.

  Making the tacit knowledge about the yet unspecified processes explicit was, of course, harder. Again, however, the template served its purposes well to both help getting started and to come up with rather uniform descriptions.

- *A Web representation of an OM makes it accessible.*

  Making the OM available in the intranet using Web technology supports its accessibility, since the users can access a central repository in a timely fashion. Still, keeping the model current requires extra effort. However, the fully automatic export facility of the RETH tool avoids the cognitive overhead otherwise involved in dealing concurrently with the model content and the technicalities of a Web representation.

- *Hyperlinks facilitate the access to definitions and can be installed easily.*

  In contrast to usual glossaries in extra documents, a glossary entry for important terms, abbreviations and acronyms is just a mouse click away through the hyperlinks. While installing such glossary links manually requires much effort, using the RETH tool allowed their automatic generation in most cases.

- *Creating the Web representation with RETH is fast and inexpensive.*

  In contrast to hand-crafting Web pages and their links as URLs, the semi-automatic link generation in and the automatic export from the RETH tool are fast and inexpensive.

- *Strict naming makes it harder but also clearer.*

  In order to make the semi-automatic link generation feasible, RETH enforces a single name space of nodes. Sometimes this makes people feel restricted, but it has the advantage of always making clear what is meant with a given term.

  There was even an additional advantage of being strict in this regard. When independently developed parts were integrated, from time to time name clashes occurred. Again, this caused problems at first glance. However, it helped finding out deeper problems with overlaps in related departments of the organization. It uncovered that they either used the same term for something different or, that they both dealt with the very same process.

- *Specialties of the development method and tool are to be hidden from end users.*

  While object-oriented modeling helped to achieve a clean structure of the OM's representation, end users were not necessarily interested in it. Therefore, the node Domain Model is renamed to Business Objects in the course of the Web export, for example. In addition, other icons for classes and instances can be used now that "hide" the object-oriented approach of development, while still keeping the clean structure.

  In particular, the experience with an early prototype showed that specialties of the modeling tool should be hidden from the end users. For example, the RETH tool requires a common anchor node for installing new nodes at the top level of the hierarchy (RETH TOP). For browsing the Web representation, it is not needed, and its original purpose was not understood by some end users. Therefore, this node is not exported to the Web representation any more.

- *End users want a usable interface to the OM.*

  In general, end users of the OM want to have reasonable usability of the related Web pages, which means that attention should be paid to aspects of human-computer interaction. For example, they want to see a relevant part of the hierarchy on the left side immediately, not just a few nodes on the top level. Similarly, they want to see some content (text or a picture) immediately and not just the structure of a node. Therefore, either the glossary text or the content of the top-most partition is shown at least and immediately on the right when a node is browsed.

- *Easy access to the content of an OM is a key issue.*

  The most important issue, however, seems to be easy access to some object in the OM, be it a document or a process description. Therefore, the entry node when getting into this Web representation is the glossary index. In contrast to the early prototype, there is string search available through an icon that is always visible on the top of each page. In addition, our approach offers yet another access path through the object-oriented hierarchy.

Ongoing Applications

Based on the positive experience with this approach, it is also used in the IST HyperKnowledge project, see http://www.verbundplan.at/HyperKnowledge/. At the time of this writing, there are several ongoing applications in two user organizations:

- Verbundplan, Austria
- Riga City Council, Latvia

Instead of using scenarios as reported above, modeling is performed in terms of *Enterprise Knowledge Patterns* (EKP) [Brash and Stirna, 1999; Zaharova and Stirna, 2000] in this project. Still, the overall approach to making an OM available in a Web representation is more or less the same.

One of the applications is modeling a business process at Verbundplan that deals with managing large unexpected outages of hydro pumps and their components (based on a case study of the Malta Power Plant). Fig. 7 illustrates part of this application. Its right side shows the entry page with explanations, which are directly embedded here. The left side shows the overall structure of this process, which primarily centers around process phases and supporting processes.

Based on the lessons learned from the previous application reported above, this Web representation hides the notion of a "Domain Model" and uses "Glossary" instead. Even a notion like "Business Object" is avoided here. In the same spirit, a folder icon is used instead of "C" for classes again. For indicating instances, a ball icon is used here instead of "I".

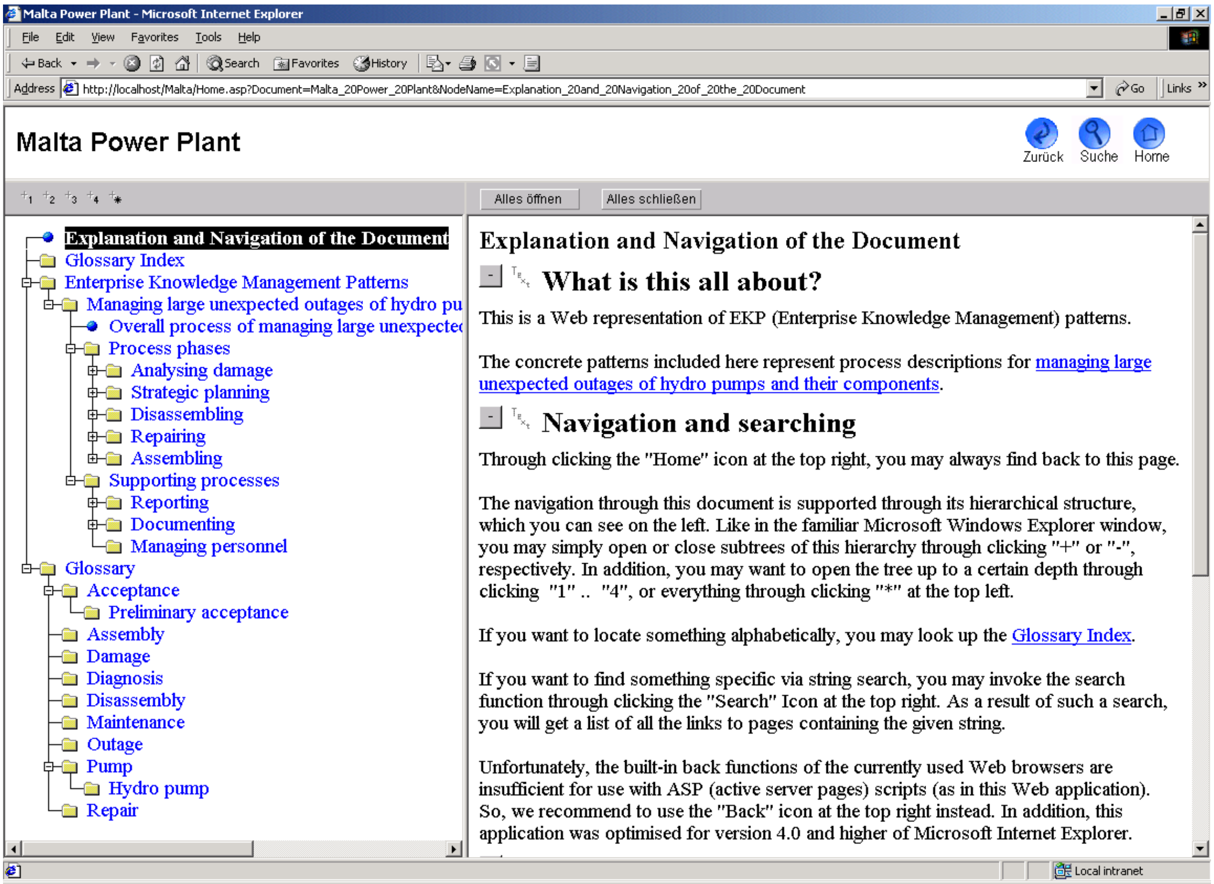

Figure 7. A screen dump of a Web-browser window illustrating a process structure.

Conclusion

In this paper, we showed how to create an intranet application that contains a special kind of business (process) model for the people that are supposed to work according to it (and not "just" for higher management and analysts). It preserves knowledge of the organization at a certain level of abstraction and provides it, e.g., for new employees not yet familiar with their daily work processes. More experienced employees still can make use of, e.g., checklists and templates in this repository.

In this sense, this intranet application serves as an organizational memory, which is represented and accessed in a timely fashion. For supporting the access to this OM, we provided standard possibilities like a glossary index or string search. In addition, the clean object-oriented structure facilitates another access path through the generalization hierarchy.

In summary, this non-standard use of a business (process) model as an OM serves corporate knowledge management. Based on this experience and the lessons learned, this approach is used at the time of this writing in several ongoing applications in two different organizations.

Acknowledgments

We would like to thank the RETH development team for all the effort and dedication to develop this tool as well as its Web export, especially Mario Hailing, Vahan Harput, Stefan Kramer and Yong Wang. We also acknowledge the hard work of those creating the Web application presented here, especially Gerhard Hackl and Sonja Winkelbauer who contributed in designing the core structure. Mario Hailing, Vahan Harput and Paul Johannesson provided useful comments to an earlier draft of this paper. The ongoing applications in the context of a trial project named HyperKnowledge are supported by the European Union under the contract number IST-2000-28401, see http://www.verbundplan.at/HyperKnowledge/. The process structure for dealing with the Malta Power Plant application was developed together with partners from Verbundplan.

References

D. Brash, and J. Stirna, Describing best business practices: A pattern-based approach for knowledge sharing, in *Proceedings of the Conference on Managing Organizational Knowledge for Strategic Advantage: The Key Role of Information Technology and Personnel (SIGCPR 1999)*, New Orleans, USA, 1999.

T.H. Davenport. Knowledge management strategies for complex organizations. *CIO Journal*, 2:55–56, July 2000.

I. Jacobson, M. Ericsson, and A. Jacobson. *The Object Advantage: Business Process Reengineering with Object Technology*, 1994, Addison-Wesley, Reading, MA.

H. Kaindl. How to identify binary relations for domain models. In *Proceedings of the Eighteenth International Conference on Software Engineering (ICSE-18)*, pages 28–36, Berlin, Germany, March 1996, IEEE.

H. Kaindl. A practical approach to combining requirements definition and object-oriented analysis. *Annals of Software Engineering*, 3:319–343, August 1997.

H. Kaindl, S. Kramer, and P.S.N. Diallo. Semiautomatic generation of glossary links: A practical solution. In *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia (Hypertext '99)*, pages 3–12, Darmstadt, Germany, February 1999.

H. Kaindl. A design process based on a model combining scenarios with goals and functions. *IEEE Transactions on Systems, Man, and Cybernetics (SMC) Part A*, 30(5):537–551, September 2000.

H. Kaindl, S. Kramer, and M. Hailing. An interactive guide through a defined modeling process. In *People and Computers XV, Joint Proc. of HCI 2001 and IHM 200*1, pages 107–124, Lille, France, September 2001. Springer-Verlag, London, England.

T.W. Malone, K. Crowston, J. Lee, B. Pentland, C. Dellarocas, G. Wyner, J. Quimby, C.S. Osborn, A. Bernstein, G. Herman, M. Klein, and E. O'Donnell. Tools for inventing organizations: Toward a handbook of organizational processes. *Management Science*, 45(3):425–443, March, 1999.

I. Nonaka, and H. Takeuchi. *The Knowledge-Creating Company*, 1995, Oxford University Press, England.

M. Polanyi. *The Tacit Dimension*, 1966, Routledge and Kegan Paul, London.

J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, Reading, MA, 1999.

S. Zaharova, and J. Stirna. Using Organisational Patterns as a Technique for Knowledge Management, *5th CAiSE/IFIP8.1 International Workshop on Evaluation of Modelling Methods in Systems Analysis and Design (EMMSAD'00)*, Keng Siau (ed.), 2000.