

# Process Oriented Knowledge Management

Stefan Jablonski, Stefan Horn, Michael Schlundt  
University of Erlangen-Nuernberg  
Computer Science Department, Chair for Database Systems  
Martensstrasse 3, D-91058 Erlangen

**Abstract.** Knowledge management and process orientation meanwhile are well established techniques. Nevertheless, their practical use still shows some severe deficiencies. In this paper we show that these shortcomings can optimally be overcome when leveraging both techniques on each other. This idea consequently leads to the idea of process oriented knowledge management. The paper presents the basic idea of process oriented knowledge management as well as a first prototype implementation.

## 1 Basic Idea

Modern economy is challenged by three requirements strongly influencing the success of an enterprise. These challenges are productivity, time-to-market and flexibility with respect to customer demands. Since economical research is not the scope of this paper, we boil down this situation into the following observation: new products with new features must be manufactured as fast as possible (at least faster than the main competitors) and at lowest costs. How can this be achieved? One solution is to deploy newest technology and most skillful experts. However, this approach might be pretty expensive since both, newest technology and most skillful experts, demand huge investments. Another approach is to exploit skills and technology that are already available in an enterprise more optimal. This implies better utilization of enterprise internal experiences and knowledge or to say it in other words the demand for an optimal reuse of technology and skills within an enterprise.

We see two main areas where the reuse of technology and skills is facilitated appropriately. As a first approach, we look into business process modeling. As a second approach we investigate knowledge management.

Business process modeling shows several facets. Firstly, it describes how business processes are performed in order to provide a template for future executions of these business processes. Thereby, their improvement is an issue permanently. Thus, experiences about how business processes are executed optimally is reused in future deployments. Secondly, business processes prescribe how given resources are optimally utilized in order to perform applications. [HaCh94] and [NiPi95] claim that analysis, documentation and modeling of business processes always has belonged to the main focus of economical efforts. One of the major goals of business process modeling has been the optimization of throughput and responsiveness of industrial applications.

A second area that aims at the reuse of experiences and skills is knowledge management. Knowledge management is a means to manage the know how and experience of employees. The objective is to treat knowledge as a valuable resource as condition for the learning enterprise [ReMS00][Nort98].

Thereby one must distinguish between information and knowledge. Knowledge must be seen as network of information and cannot be stored in databases or documents; information is contained in these carriers. Knowledge must be compiled out of information according to a specific application context. Information can be accumulated in a large repository in order to place it at disposal. Knowledge management supports a user by offering appropriate information to build up the knowledge he needs in a certain problem solution process. Usually, information is contained in documents of various formats. The collection of documents then constitutes the knowledge base. As an example application, a draftsman's work behavior is analyzed. When he has to cope with a serious technological problem, he consults the knowledge base in order to obtain information about solutions

to similar design problems. According to [Vers00] enterprises expect various benefits by deploying knowledge management (cf. Fig. 1)

We favor both approaches, business process modeling and knowledge management, to facilitate reuse. Nonetheless, we see that both approaches also show drawbacks. The major obstacle of business process modeling is that knowledge which is not directly associated with a concrete business process, cannot be handled properly. A severe flaw of knowledge management is the missing structure that particularly facilitates easy access and use of the knowledge base.

Our idea is to combine both approaches, business process modeling and knowledge management. Both concepts should leverage on each other in order to eliminate their drawbacks identified above. To be more precise, this means that we utilize business processes as an ordering dimension in the knowledge base. Nevertheless, the knowledge base additionally offers further ordering dimensions than business processes in order to cope with knowledge that cannot be directly associated with business processes or parts of it. We characterize our approach as process oriented knowledge management.

Section 2 introduces the two basic concepts of our approach, knowledge management and business process modeling. Section 3 then presents the concept of process oriented knowledge management.

Implementation issues are discussed in Section 4. Hereby we fall back on experiences we made with a

prototype that has been developed within the scope of two industrial projects. A conclusion (Section 4.2) summarizes the main contributions of this paper and presents experiences with our prototype implementation.

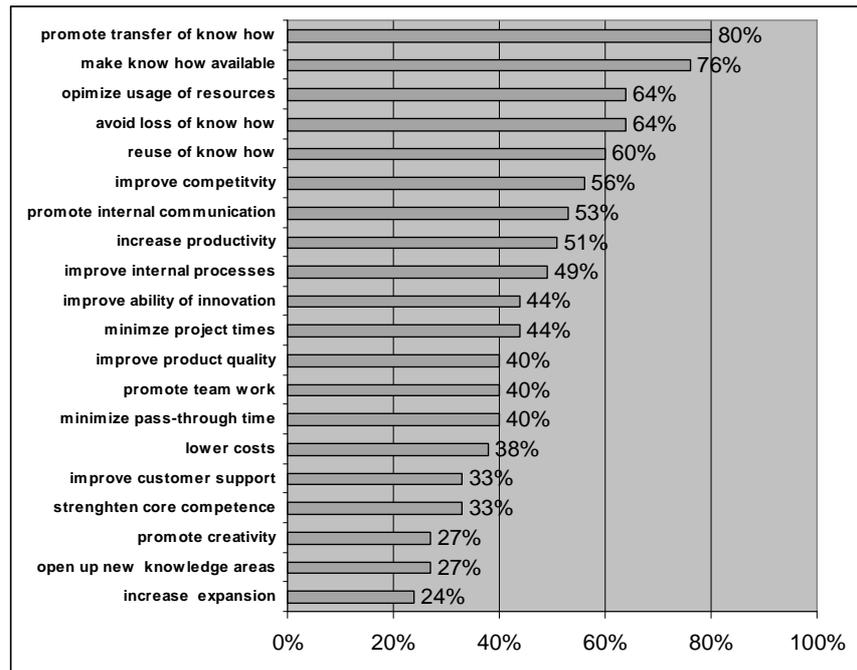


Fig. 1: Positive Effects of Deploying Knowledge Management [Vers00]

## 2 Knowledge Management and Process Orientation

This section presents the two basic concepts the paper is based on: knowledge management and process orientation. Besides, it is shown how these technologies are enacted. Finally, open issues are revealed substantiating a need for an enhanced concept that leverages the two concepts knowledge management and process orientation.

### 2.1 Basic Concepts of Knowledge Management

First, an example scenario is shown where the need for knowledge management is justified. After that, the basic ideas of knowledge management are analyzed. These contributions finally lead to definitions of the terms knowledge, knowledge management and knowledge carrier.

As an example consider a draftsman who has to dimension a steam pump for a certain aggregate in a large power plant project. Many constraints concurrently have to be taken into account depending on

technical and environmental issues of the aggregate and the whole project. The dimensioning process becomes very complex and difficult. Because of this complexity, the draftsman is continuously facing new situations. Nevertheless, he will establish a certain set of experiences through his career which make him continuously improve his work over time. This means that the knowledge of the draftsman increases. He naturally re-uses parts of his knowledge in new challenging situations. This knowledge is often called experience or know-how.

Re-using knowledge forms the basic idea in knowledge management. Another basic idea is to share knowledge. The latter requires to publish knowledge in such a way that other people can consume it. This facilitates re-use not only by one person that has generated knowledge but also by other persons that are able to share the knowledge. In order to enact knowledge management the following tasks have to be solved [ReMS00]:

- Identification (Where can I get information?)
- Acquisition (How can I get the information?)
- Structuring (How to organize information?)
- Promotion (How to stimulate new ideas?)
- Storage (Where should the information be stored?)
- Distribution (How to bring information to users?)

In the scope of this paper we don't deal with information "Identification"; this task needs to be performed by experienced users of a knowledge management system, eventually supported by tools (e.g. data mining tools). Neither we want to consider information "Promotion". Knowledge promotion is a psychological task for efficiently using and expanding the enterprise wide knowledge.

Our main focus lies on information "Structuring" and "Storage" because especially for the first point we see the most severe flaws with conventional tools. When investigating "structuring" in the realm of knowledge management two issues arise. First, information has to be prepared in such a way that it can be shared. Second, information has to be organized in such a way that it can be found when it is needed. The first issue requires for a suitable form of representation. The second issue directly points to a classifying structure where information can be put into.

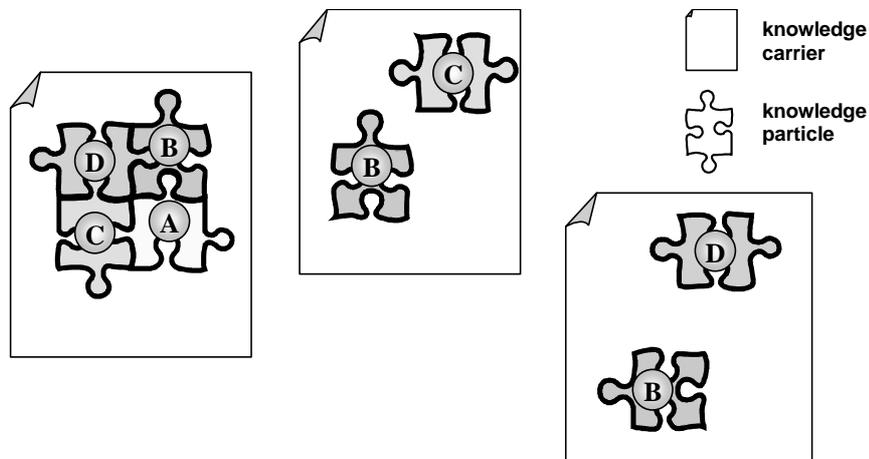
Information is widely spread over most parts of an enterprise. In this paper, we concentrate on information which is represented on a computer system. This implies that information "acquisition" is in our context interpreted as having access to information via a computer network. Here, knowledge is not directly represented and stored, but knowledge is part of document files, databases and other computer based media. In this paper we call these media *knowledge carriers*. Knowledge carriers are of different types. Knowledge contained in knowledge carriers are from now on called *knowledge particles*. A *knowledge base* is the repository that gathers knowledge.

There are two ways to build up a knowledge base. First, information is extracted out of the knowledge carriers, is converted to the proprietary format of a knowledge base and is classified into the knowledge base. This requires a large amount of work to prepare - i.e. to extract, to convert and to arrange - knowledge for later use. Nevertheless, this approach is pure from a logical point of view since the knowledge base is fed with knowledge, i.e. knowledge particles. Knowledge particles are intensional representations of knowledge.

A second approach organizes the knowledge carriers that keep the knowledge particles. In contrast to the intensional nature of knowledge particles, knowledge carriers bear an extensional character. They represent the tangible representations of knowledge. With this second approach, the preparation work, i.e. information extraction and conversion, can be neglected to a great amount. Complete knowledge carriers are stored in the knowledge base. Since the preparation work forms huge overhead, we favor the second approach.

Both approaches require to structure the knowledge base. This is necessary in order to later find knowledge, i.e. knowledge particles. We call this ordering structure the *schema (or structure) of the*

*knowledge base*. We regard this schema as very valuable and therefore also identify it as particular knowledge. The schema of a knowledge base is formed according to individual criteria of an application area.



**Fig. 2: Knowledge Particles and Knowledge Carriers**

Yet we do not want to conceal the drawbacks beard in our favored approach. Normally, each knowledge carrier will contain different pieces of information, i.e. different knowledge particles. For instance, a design document – as the knowledge carrier - not only stores knowledge particles associated with the design drawing itself (e.g. geometric data) but also holds information concerning the design process, for example the name of the designer and the date of the last modification of the design drawing. On the other hand, a particular knowledge particle can be stored on different knowledge carriers. For example, the name of the designer of a specific part not only is contained in the design document but also is kept in a workflow history. Fig. 2 depicts the n:m-relationship between knowledge particles and knowledge carriers.

Nevertheless, the drawback formulated in the former paragraph also bears an advantage. The knowledge carriers form the context for a knowledge particle in a natural way. For example, an address is instantaneously recognized as receiver of some goods. This context information would have to be extracted additionally in the first approach of knowledge management introduced before.

## 2.2 How Knowledge Management is Facilitated

This chapter briefly discusses existing concepts which are used to handle the discussed knowledge management issues:

*Artificial intelligence.* Information is extracted from knowledge carriers and is stored in a knowledge base. A knowledge base has a propriety format, information has to be managed so that it can be queried and accessed later on [GeNi88], [Fros86]. The artificial intelligence approach starts from the observation that a knowledge base is separated from the knowledge carriers. This necessitates information extraction and presentation as an additional task in knowledge management. The knowledge base must be structured in a way such that users can easily orientate themselves in the knowledge base. Due to the separation of the knowledge base into an independent component, the knowledge base is not really integrated into an application system. Thus, it will not be used as often as it would be necessary. Besides, due to the propriety format of knowledge representation in the knowledge base, the ease of access for a common user is questionable.

*Document Management:* Document Management Systems enable storage of different document types within one central database. Functionalities for enriching these documents by meta information and finding documents are supported. Also in document management system, a schema is necessary in order to provide a structure documents might be classified into.

*World Wide Web.* One intention of the world wide web is to offer information in a broad manner for everybody. Using the terminology of the previous sub-section this can be interpreted as offering knowledge carriers on the world wide web. These knowledge carriers remain uninterpreted. They are mainly managed by those people who made the knowledge carriers available. One advantageous property of this method is that knowledge carriers remain unchanged and additional extraction need not to be done. One drawback of the world wide web is that preferably HTML documents are handled. Also, the world wide web would need a structure in order to provide better orientation when surfing on the web. On the one hand, HTML documents can easily be linked such that a structure is simulated. On the other hand, the missing structure is compensated by search engines who aim at the provision of direct access to documents independent from their location. Although, search engine can only be regarded as partial solution. The selectivity of search queries is often so bad that a huge, not comprehensible number of documents is returned. This flood of documents makes it impossible to find the information needed. Without any doubt, the main advantage of the world wide web is its availability and ubiquity.

Decisively for the acceptance of knowledge management the meaningfulness of the schema of the knowledge base. If it is good, the knowledge base will be used; if it is not comprehensible, the knowledge base will just be neglected.

### **2.3 Basic Concepts of Process Orientation**

Analysis, documentation and modeling of business processes (we prefer the term application processes in order to include either business oriented, administrative and technical processes) have always been performed despite the advent of workflow management. Nevertheless, workflow management has emphasized the need for process orientation. One goal of process orientation is to reach a global optimum of a comprehensive application system. Function oriented strategies in contrast end up in a couple of local optima which not necessarily sum up to a global optimum [HaCh94] [NiPi95].

Among other features, comprehensiveness best characterizes process orientation. Whereby comprehensiveness shows at least two facets. A first facets relates to the contexts of process description: they ought to be complete, i.e. they either comprehend process steps, control and data flows, organizational structures, application systems and further relevant ingredients of a process [JaBu96]. A second facet points to the realm a process encompasses. It is not limited by organizational or technical boundaries within an enterprise but is content related and therefore embraces all relevant components independent of the organizational or technical section they belong to.

A most important issue of process description is to draw the line between components of an application system being still relevant for a process and components which are not further of interest for a process. It is problematic to anticipate this interest either because it is not known what information will be necessary when process will be performed and because it is not known what kind of information could become available in the future. Thus, the description of an application process is always limited and often situations will occur where relevant information cannot be associated with the application process although this information would be most valuable.

### **2.4 How Process Orientation is Facilitated**

There are two major areas of process enactment. The first area is limited to the sophisticated description of application process, it is called business process modeling [Sche98]. The goal is to deliver a comprehensive description of real world processes and – for instance – to simulate its behavior in order to get feedback for improvements. The second area extends application process modeling to process enactment. Two predominant representatives of this area are workflow management [JaBu96] and groupware [ElGR91]. The former enacts well structured application processes whereby the latter focuses on highly dynamic and unstructured application processes.

It is not in the scope of this contribution to discuss either business process modeling, workflow management and groupware in greater detail. Nevertheless, some short assessment of the state of the art of these technologies should be conveyed. The main drawback of most approaches in all three areas

is the lacking flexibility with respect to extensibility and openness. It is more or less impossible to extend a model of one of the three areas in such a way that some more information elements can be associated with a specific part of a process description. So, it is not possible to add more information to either a business process, a workflow or a groupware model in case it is needed. This behavior naturally results in incomplete descriptions of real world scenarios. The missing information often leads to sub-optimal enactment of application processes.

## **2.5 Open Issues**

We have seen that each of the enactments of either knowledge management and process orientation are limited with respect to ease of use and effectiveness. Our analysis of both areas also has revealed that there is a great potential to leverage on each other. This means in detail:

- The missing structure of knowledge bases can be compensated by adopting process description as one specific foundation for the schema of the knowledge base. This policy alleviates the access of a knowledge base since in process oriented application systems the user knows exactly his position in the application process and can therefore easily navigate into the knowledge base.
- The missing extensibility and openness of process oriented concepts can be compensated by linking directly to a knowledge base from a process description. This is possible since process description themselves function as structuring criteria of a knowledge base. At the corresponding spot in the knowledge base, all relevant information of a process step – that goes beyond the capability of a process model – can be captured.

We are convinced that the two concepts process orientation and knowledge management ideally complement themselves. The synergetic effect of integrating the two concepts forms a powerful tool-set to enact large, knowledge intensive applications. In the next two sections, we present a concept that leverages on process orientation and knowledge management and integrates them to a persuasive means.

## **3 Process Oriented Knowledge Management**

This chapter describes how knowledge management and process orientation can be combined and what issues have to be considered hereby. These issues are separated into functional and non-functional ones.

### **3.1 Basic Issues**

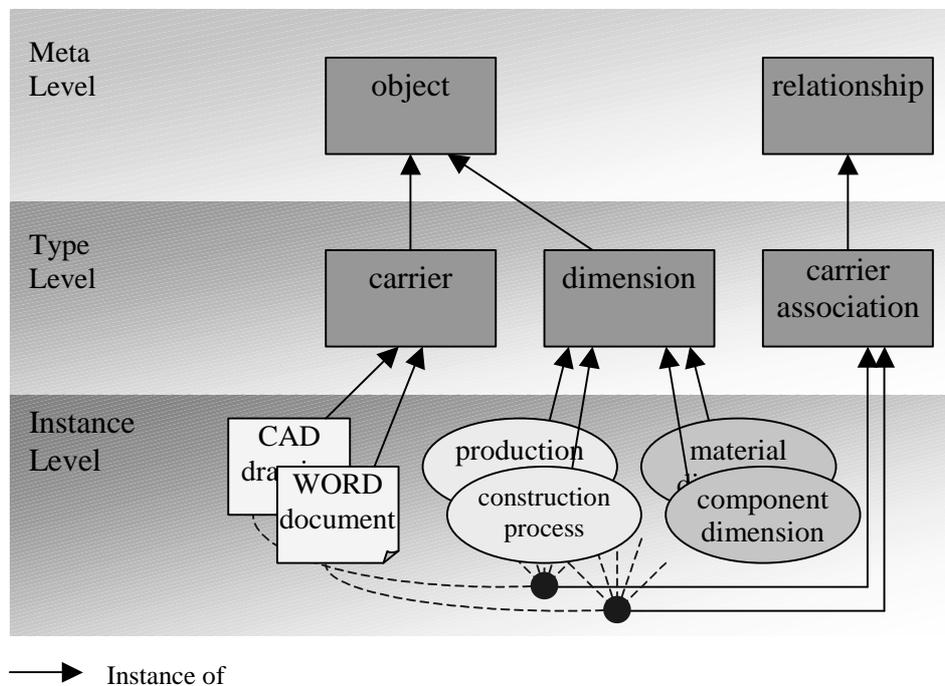
Two main concepts form the basic idea of process oriented knowledge management. On the one hand all knowledge particles (cf. Section 2.1) of an enterprise have to be classified according to a sophisticated set of criteria. On the other hand it is crucial to pursue an open enactment strategy. Both concepts are discussed in the following.

As mentioned before knowledge itself cannot be stored or computed but knowledge can be compiled out of a network of information. Information is regarded as knowledge particles stored on knowledge carriers. Since we primarily deal with knowledge carriers they are viewed as first class citizens; in contrast, knowledge particles are classified as second class citizens. To define a schema for a knowledge base means to specify a structure where knowledge carriers have to be put into. Information structuring which finally results in classifying knowledge carriers is achieved as follows.

1. The ordering dimensions of a knowledge base are defined. Dimensions are defined freely by knowledge administrators and can be extended arbitrarily. Generally, such a network complies to the metaphor of associative connections of information. This mechanism is similar to how a human brain stores and retrieves information. This main idea is described in more detail in Section 3.2.

2. Knowledge carriers are connected to a set of dimensions that best characterize their contents (i.e. knowledge). One particular dimension of a knowledge base shall be formed by processes.

The dimensions of a knowledge base must ideally fit to physical (e.g. organizational units) or logical (e.g. business processes) structures of an enterprise. This would drastically increase its acceptance and familiarity.



**Fig. 3: The Tree Level architecture**

Openness represents another main requirement towards a knowledge base. Openness means that arbitrary objects (i.e. knowledge carriers) and arbitrary classification structure can be used (cf. Fig. 3). This feature enables the adoption of a knowledge base to an arbitrary application field. We achieve openness by defining the contents of a knowledge base on a meta object level. Here, just the two entities “object” and “relationship” are known. They are instantiated on the type level: “carrier” and “dimension” are derived from “object”, “carrier association” is derived from “relationship”. A carrier is linked to one or more dimensions by a carrier association. Since dimension is a complex object (it is a sequence of values), a carrier is linked to a value of a dimension in order to be more precise. Now, each application area is free to instantiate the terms “carrier”, “dimension” and “carrier association” at its own need. This feature facilitates openness. For example, a manufacturer would define the dimensions “material” and “part list”, while a sales department would prefer to have dimensions like “customer” and “product line”. Furthermore, the manufacturer would instantiate the carriers “CAD drawing” and “NC program”, while the sales department would go with the carriers “WORD document” and “Spread sheet”. Note, each instantiation has to be done individually and is exclusively justified by a need of a specific application area.

## 3.2 Functional Constituents

This section details the functional constituents of a knowledge management system. As already discussed, knowledge carriers and dimension have to be regarded at least

### 3.2.1 Knowledge Carriers

In our approach information is represented on knowledge carriers. The knowledge carriers are arbitrary documents which might contain any kind of information. For instance a simple text

document is a knowledge carrier. More complex examples are the representation of a specific business process in a PDF document, an image file or a CAD drawing.

The knowledge management system must provide functionality to find problem relevant knowledge carriers and finally provide access to the knowledge particles stored on different knowledge carrier. Therefore, knowledge carriers must show several features.

Firstly, knowledge carriers need to be tagged with attributes. Attributes are important for the retrieval of the knowledge carrier. A very common way to select knowledge carriers out of the knowledge base is to search for knowledge carriers with a certain set of attribute values. The main benefit of using attributes is to be able to identify knowledge carriers without having to analyze their real contents. Just think of how complicated and cumbersome – or even impossible - it is to search for a knowledge carrier with a specific image. Without attributes this knowledge carrier could only be found by sequentially browsing through the knowledge base.

Moreover, attributes classify knowledge carriers independently of any dimension. This is important because some kinds of knowledge carrier just cannot be associated with a certain dimension. For instance, the examples at the beginning of this sub-section do not show places in the dimensions where the author of a document or its date of creation could be noted elegantly.

Another important issue is how to store knowledge carriers in the knowledge base. The knowledge base system must be able to either store knowledge carriers in a separated database or to handle pointer references to external knowledge carriers. In the first case all important and relevant information of the knowledge base is held by the knowledge management system; in the second case only meta-attributes and information structure are held by the knowledge management system. This bears the danger that external knowledge carriers are out of the scope of control of the knowledge management system. Nevertheless, this approach also is disputable since it avoid huge redundancy which would be introduced when all often knowledge carriers are copied into the knowledge base.

### 3.2.2 Dimensions

Dimensions are used in order to categorize knowledge particles residing in knowledge carriers. They organize knowledge particles (intensionally) by the mean of finding knowledge carriers (extensionally) that contain these knowledge particles (cf. Section 2.1). Thus each dimension contributes a part of the classification of a knowledge carrier.

We define dimensions by the following formalism:

Let  $ET$  be a set of entry types.  $ET$  determines all possible entries a dimension encompasses. An occurrence  $D_i$  of a dimension is primary formed by a subset of these  $ET$ .

$$(1) \quad D_i = \{ e \mid e \in ET \}$$

The knowledge carriers are now associated to  $n$ -tuples of  $D_i$ . Such an  $n$ -tuple identifies a knowledge carrier within a dimension. The cardinality of this association between knowledge carrier and  $n$ -tuple is  $m:n$ , which means one knowledge carrier may be associated to many tuples or one tuple may be associated to many knowledge carriers. For each  $n$ -tuple the parameter  $n$  can be chosen arbitrarily. One particular tuple  $t_i^n$  then identifies a combination of values of  $e_j$  at specific positions where at least one knowledge carrier is associated within the dimension  $D_i$ .

$$(2) \quad t_i^n = \langle e_1, \dots, e_n \rangle \mid e_j \in D_i$$

The set  $T_{all\_kc}$  over  $n$ -tuples summarizes all  $n$ -tuples used as associations within the dimension  $D_i$  (whereby  $m$  is the number of associated knowledge carriers).

$$(3) \quad T_{all\_kc_i} = \{ t_i^n \}$$

In general the possible solution space for a knowledge carrier is the  $m$ -fold cartesian product  $T_{all}$  over all entry types in  $D_i$ .

$$(4) \quad T_{all_i} = \times_{j=1..m} D_j; \quad m = \max\{n \mid \exists t_i^n\}$$

Concluding the following expression holds:

$$(5) \quad T_{all\_kc_i} \subseteq T_{all_i}$$

For practical reasons it is reasonable to restrict this solution space according to application pragmatics. Therefore one may define any ordering structure on top of the entries in  $D_i$ . A bill of material for example sets up a hierarchical structure over the entries, which means that one entry may not contain itself. This can be formally expressed by

$$T_{all\_assembling} = \{ \langle e_1, \dots, e_n \rangle \mid \forall i, j \leq n \wedge j > i: e_i \neq e_j \}.$$

Generally a half ordering is established over  $D_i$  using a relation  $R$ . In the above example  $R$  is the “contained-in” relation. We want to stress that  $R$  is only an example for such an ordering structure, in general the ordering structure may be chosen arbitrarily. The identification of knowledge carriers is not delimited by any  $R$ , but  $R$  may be a benefit for knowledge carrier retrieval, if users agree on the structure of the dimension  $D_i$  formed by  $R$ .

Let  $R$  be a relation on  $D_i$ :  $R$  is a half ordering relation when

$$(6.1) \quad \forall e \in D_i : e R e \quad \text{reflexivity}$$

$$(6.2) \quad \forall e, f, g \in D_i : e R f \wedge f R g \Rightarrow e R g \quad \text{transitivity}$$

$$(6.3) \quad \forall e, f \in D_i : e R f \wedge f R e \Rightarrow e = f \quad \text{antisymmetry}$$

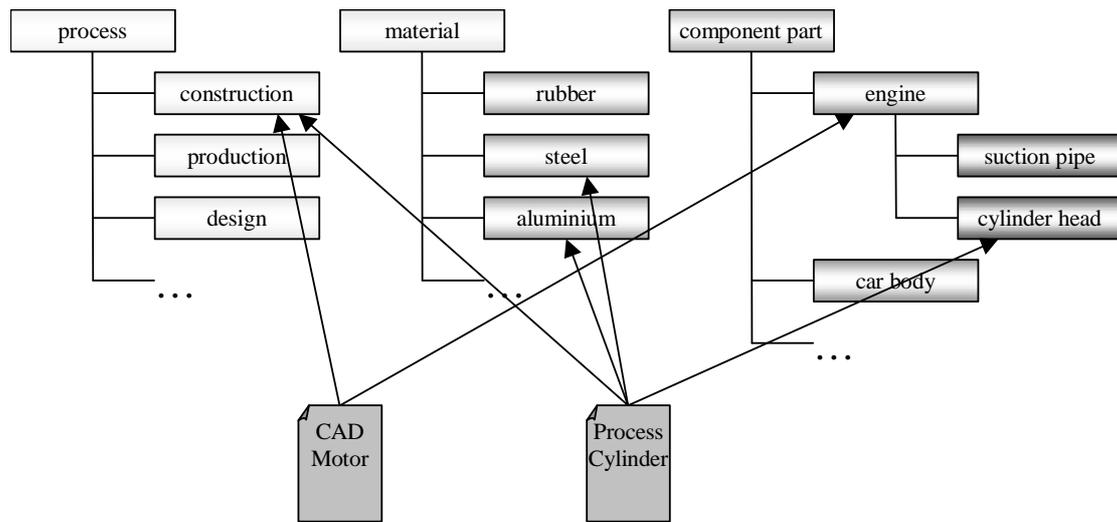
A second example for restricting  $T_{all}$  uses the tuple position of an entry as some priority. Different knowledge carriers can for example be weighted by size, age and security level. These entries are prioritized according to their positions in the association tuple. In this example the entries can occur at any position but all have to be different. This can be formally expressed by

$$T_{all\_priority} = \{ \langle e_1, \dots, e_n \rangle \mid \forall i, j \leq n \wedge j \neq i: e_i \neq e_j \}.$$

This example can not be expressed by a half ordering relation on top of  $D_i$  as the property of antisymmetry (6.3) does not hold.

Finally many dimensions may be used in one application context. Each of these dimensions is constructed as explained above. If ordering relations are used, these need not to be the same over all dimensions. If many dimensions are used a knowledge carrier can be associated not only with one dimension by one tuple, but with  $m$  dimensions by  $m$  tuples. Further more several different associations per knowledge carrier can be defined. Searching knowledge carriers may now be done by searching the knowledge carrier within one dimension and connecting all dimensions together by using set operations.

In the following we describe an example related to an industrial context. The categorization of knowledge carriers is done by domain experts. Several independent criteria (i.e. dimension) are available simultaneously. Furthermore each dimension is here hierarchically decomposed. The here used term of dimension is comparable to the dimensions used in the data warehousing area [Kimb96], [ChUm97].



**Fig. 4: Information Structure**

In Fig. 4 an example information structure in the area of automobile production is sketched. There are three main dimensions “Process”, “Material” and “Component Part”. These dimensions are partly hierarchical decomposed. This decomposition may be extended during runtime of the knowledge management system. Furthermore, the knowledge management system organizes the actual information network (pointers in Fig. 4). These pointers associate the knowledge carriers and by that the contained knowledge particles to on ore more values of the dimensions. For instance, the knowledge carrier “CAD engine” is associated to “Process>Construction” and “Component part>Motor”.

Associations can be added or deleted according to the users’ interpretation. On purpose these associations may even be redundant or even contradictory. This is why links reflects the knowledge maps within the users’ brains. Finally, this network is ought to reflect a good compromise of the knowledge of participating users. Non experienced users may utilize the given associations to get access to desired knowledge particles via knowledge carriers. Thus, the associations are used as a navigation mean, too.

In contrast to data warehouses we claim that a knowledge carrier does not need to be assigned to every dimension. In other words the dimensional categorization of a knowledge carrier is optional. This technique increases the user’s flexibility and facilitates to extent the structure.

Our experiences in industrial projects (at Volkswagen AG, Siemens AG, DaimlerChrysler AG) have shown that most knowledge carriers originate from experiences the employees gain when executing specific steps of an application process. In order to take this experience into account our approach uses processes as a specific dimension. Processes form a global structuring criterion that can be found in each business application. Since not every knowledge carrier has to be assigned to the process dimension this approach cannot be called process centric, rather we call it process oriented. The following characteristics of process orientation can be recognized:

- Processes for a suitable dimension

When processes are used as a knowledge dimension, they act as integrating constructs. This is applicable since integration is one of the main issues of process orientation. Along with business process analysis many knowledge carriers are assigned to different process phases within the process model anyway.

- New interrelationships between knowledge carriers become obvious

Because of the integrating character of processes, new process oriented interrelationships between different knowledge carriers become obvious.

- New comparison criteria between different knowledge carriers emerge

Process oriented comparison criteria come into focus, like e.g. related steps, responsible person, process contribution, etc.

Process oriented knowledge management only uses processes as structuring criteria. The administration of the process models has to be performed externally.

### 3.3 The Non-Functional Constituents

In contrast to the functional requirements, the non-functional requirements mainly concern the usage of the knowledge management system. We don't want to delve all these points in detail, since these requirements are common in modern system environments. We only want to stress issues concerning our approach of knowledge management.

- *Ubiquitous access* means that any participant user must be able to access the knowledge base from any system in the enterprise, i.e. the intra net. That implies that the search front end of the knowledge management system should be realized as browser-enacted application to overcome heterogeneous system environments.
- *Openness* addresses the need for adaptations. As not all needs of the application scenario can be anticipated the knowledge management system must be extensible by additional features. This extensibility concerns knowledge carriers, information structure and also the knowledge management system itself.
- *Availability* means that the knowledge carriers which are managed by the knowledge management system must be accessible from any place at any time by every authorized person. This omnipresence is crucial for the wide use of the knowledge management system. This issue is demanding, when external referenced knowledge carriers were used.
- *Usability* is essential for a system that should be deployed by many employees of an enterprise. The most demanding issue here is the trade off between mighty functionality (often implies complex use) and clarity (often implies restricted functionality).

Knowledge management can only succeed if as many involved people participate as possible because each participant holds and generates knowledge. The more information the system can provide the more users can be supported by the system. But it must clearly stated, that a technical solution for information distribution is not sufficient for knowledge management. Successful knowledge management is based upon an enterprise culture, where every employee is prepared to share his knowledge and where everybody catches the grasp for knowledge management.

## 4 Implementation of a Process Oriented Knowledge Management System

This section gives a short overview on implementation issues of a process oriented knowledge management system. First, we explain the system architecture of our prototype: participating components and their interrelationships are described. The subsequent sub-section shows how this architecture fulfils the requirements sketched in Section 3. Finally, a kind of a handbook is delivered that describes how to use our process oriented knowledge management system.

### 4.1 Prototype System Architecture

Fig. 5 depicts the system architecture of our prototype. The system's core component is the Know How Manager. The Process Interface is used to feed processes into the Know How Manager; these processes build a dimension of the knowledge base. Knowledge carriers are brought into the

knowledge base – either physically or by reference - through the Carrier Interface. Either dimensions, knowledge carriers and attributes associated with knowledge carriers are stored in a database. A world wide web (WWW) front end provides ubiquitous access to the knowledge base.

Knowledge carriers can emerge from different sources as the world wide web, a file system or a document management system. Extensibility of the document interface allows to adapt to new sources. In principles, knowledge carriers could be copied into the knowledge base; alternatively knowledge carriers could also be made known to the knowledge base by providing a pointer to its actual position outside the process oriented knowledge base.

Through the Process Interface a process structure is fed into the system; it is used as dimension for the knowledge base. Since only structural information of processes is needed the Process Interface can easily be adopted in order to connect to arbitrary process modelers. An internal representation of a process trees is stored in the database of the knowledge base. Coherence between this internal representation of a process structure and the external, original process structure has to be ensured over time. Changes in the external process structure might cause comprehensive re-assignments in the knowledge base. It is out of the scope of this paper to discuss this problematic administration task.

A WWW front end builds the user interface to the knowledge base. In our architecture the know how manager acts as server whereas the users' front end components play the client role. Many instances of the front end component might access the know how manager concurrently and thus have to be synchronized. The WWW front end also incorporates a special administration interface. Besides other things, this interface is used to define dimensions of the knowledge base in addition to the process dimension which is gained through the specialized Process Interface.

It is evident that the prototype implementation copes with the functional requirements posted in Section 3.2. Either knowledge carriers and dimensions are handled properly while the particular dimension “process structure” is treated separately. The non functional requirements (cf. Section 3.3) ubiquity, availability, usability and extensibility are mainly handled by the WWW front end. As the world wide web is accessible from nearly all over the world ubiquity is achieved. The know how manager ensures the availability of all managed knowledge carriers. The client-server relationship between front end and know how manager thus provides availability of all knowledge carriers to the front end users.

As the WWW is widely established most of the users will be familiar with the utilization of such front ends. Thus, usability is achieved as long as the design of the front end is intuitive and easy to understand. Extensibility is mainly guaranteed by the data schema introduced in Section 3.1: the meta level approach allows arbitrary configuration of either dimensions and types of knowledge carriers.

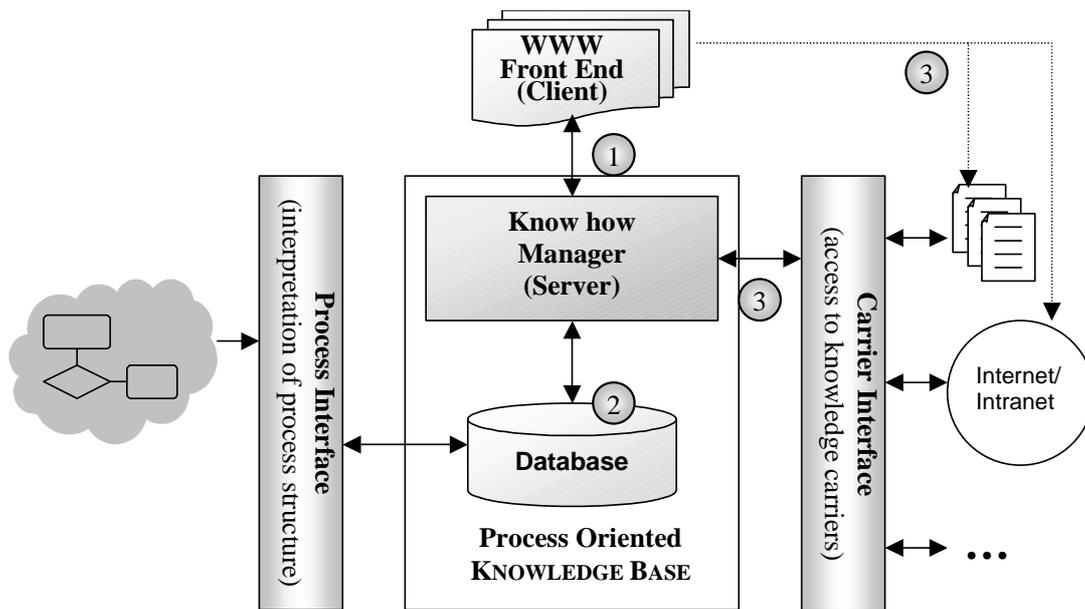


Fig. 5: Prototype System Architecture

## 4.2 Implementation Issues

This section briefly describes how the proposed system architecture (cf. Section 4.1, Fig. 5) has been implemented in our prototype. At first we show how communication between the front end components to the knowledge base is implemented. Subsequently we take a close look at the database issues (database scheme design and database communication). Finally we propose how to enact ubiquitous access to knowledge carriers.

### 4.2.1 Client/Server Components

In this section the design of the client and the server components as well as communication issues are discussed (see also Fig. 5, ①).

According to Section 3.3 ubiquitous access to the knowledge management system has to be provided. This implies that the client component has to be implemented platform independently. Further more the client component must be easily accessible, e.g. without extra installation effort. In our approach we use a Java applet as client. This applet is embedded into a HTML page and invoked within a web browser (e.g. Netscape, Internet Explorer, Opera). Thus no preliminary installation by the user is needed. One disadvantage of applets is their low performance when loading the class code into the browser. As a result we need to leave the client's program logic rather simple, so that this effect would not become too significant.

In order to make the client component as thin as possible, the complex parts of the program logic have to reside within the server component. But the more program logic is taken over by the server component the more communication between client and server has to take place. As communication between Java programs is very flexible our approach deploys a Java servlet as server component. This servlet is executed under control of the same web server process which also provides the HTML pages containing the client applet. This is essential as Java applets only may communicate to that computer where they have been downloaded from (sandbox principle [HoCo97]). Now e.g. a HTTP tunneled communication between server and client can be set up. In our implementation server and client communicate via SSL, which is a more secure way of data exchange between server and client because of the encryption of the communication channel.

## 4.2.2 Database

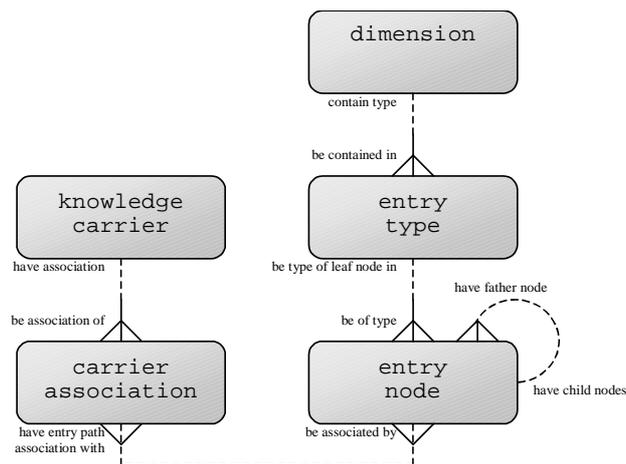
In this section the database design and the connection to the know how manager are discussed (see also Fig. 5, ②).

The flexibility and categorizing capabilities which are proposed in Section 3.2.2 are reflected by the database scheme. In Section 3.3 extensibility is identified as crucial issue for flexibility. This means that the database scheme has to be designed on a meta level. Fig. 6 shows the meta level database entities of our realization: `dimension`, `entry type`, `entry node`, `carrier association` and `knowledge carrier`. This design provides the required extensibility because instances of meta objects describe application types. New application types are added by creating new instances of the meta objects.

In Fig. 6 the interrelationships between the entities are symbolized by dotted or solid connection lines. These connections can be read in both directions. If the start point of a line is solid, a mandatory association is symbolized; an optional association is symbolized by a dotted start point respectively. A branched end point of a line expresses a “one-to-many” relationship between the connected entities while a single end point stands for a “one-to-one” relationship.

As depicted in Fig. 6 a dimension can contain several entry types, which basically are the values a knowledge carrier can be associated with. But as described in Section 3.2.2 a knowledge carrier rather is associated with a complete entry path than a single entry type. In order to express this, in the database scheme a tree like structure is set up with the entity `entry node`: one entry node can have several child nodes but at most one father node. Thus a single object `entry node` represents exactly one node within the tree of possible combinations of entry type associations in a dimension. Concluding a `carrier association` within one dimension can relate to an `entry node` which of course must be instances of exactly one entry type. One `carrier association` can relate to several entry nodes of different dimensions. Finally a `knowledge carrier` can have several carrier associations.

This scheme implements the flexibility proposed in Section 3.3, because any possible combination of entry types can be expressed by exact one entry node. If the possible combinations have to be restricted by an application, the corresponding entry nodes are left out of the database so that documents can not reference them.



**Fig. 6: Entity Relationship Scheme for Generic Knowledge Carrier Association**

To connect the database to the know how manager (cf. Section 4.2.1) we use JDBC. JDBC is widely established and available for the most database management systems. As the know how manager is implemented in Java, JDBC is the easiest way of communicating to a database. By simply using different database connect strings different databases on different computers can be addressed. Thus the system load can easily be balanced over several computers.

### 4.2.3 Knowledge Carrier Access

In this section the access to knowledge carriers is shown (see also Fig. 5, ③). Here it is necessary to provide easy access on the one hand and to ensure security issues on the other hand.

As the client component is implemented as Java applet (cf. Section 4.2.1) which is invoked by the web browser of the end user it is reasonable to provide access to knowledge carriers over the browser capabilities as well. This means that knowledge carriers are associated with an URL within the internet or the intranet. These URL have to be maintained by the know how manager component.

Knowledge carriers either are managed by the know how manager directly or they are linked and categorized by the know how manager. In the first case knowledge carriers have to be copied to the know how manager so that they can be accessed by others. When the original knowledge carrier changes the new version has to be uploaded again. In the latter case the know how manager does not directly manage the knowledge carrier. Here an external link is categorized and inserted into the know how manager database as an URL. When the document changes no further adaptations have to be made. But when the URL of the document changes the know how manager needs to be updated accordingly.

The presented document handling mechanism allows to manage all kinds of file based knowledge carriers. As the knowledge carrier access is handled over the internet or the intranet respectively security issues also apply as long as they do not exceed the capabilities of the internet or the intranet.

### 4.3 Using the Process Oriented Knowledge Management System

This section provides a brief introduction into the handling of our prototype. A first example of usage is to search for of a specific, suitable knowledge carrier. On the WWW front end the user has two alternatives to go: on the one hand, the user can perform a full-text search against the knowledge carrier attributes. This option might be suitable when the user has no glue of where to find this problem specific information. On the other hand, the user can navigate through the dimensions of the process oriented knowledge management system and browse documents which are associated to the current selections on the multiple dimensions. This approach is most suitable when the user is currently executing a specific process step and wants to find situation related knowledge carriers.

A second example shows how a new knowledge carrier is inserted into the knowledge base. In this case the WWW front end allows to upload arbitrary files or external links into the process oriented knowledge management system. First, the user selects values for relevant dimensions which best characterizing the new knowledge carrier, better the information which is contained in the knowledge carrier. After that, the upload procedure is performed. The association to the selected dimension values is done automatically by the know how manager. The uploaded knowledge carrier can then additionally be associated with attributes that further characterize it.

## 5 Conclusion and First Experiences

This paper demonstrates how knowledge management and process orientation can be integrated in a synergetic manner. Process oriented knowledge management as the result of this integration effort eliminates some of the major drawbacks of the two basic techniques underlying this concept. Our first experiences in a project with one of the larges German car manufactures already shows positive feedback by the users. Due to the introduction of the process dimension into the knowledge base they report of an increased familiarity with the knowledge base and of a increased readiness to make use of it. On the other hand side they appreciate the conceptual enhancement of pure process modeling since additional information about processes can ideally be stored in the process oriented knowledge base.

Due to the positive feedback of our prototype we aim at a close integration of a workflow management system into the architecture depicted in Fig. 5. This means the possibility to directly link into the process oriented knowledge base from the work-list of a workflow management system.

## References

- ChUm97 Chaudhuri, S.; Umeshwar, D.: An Overview of Data Warehousing and OLAP Technology. SIGMOD Record 26(1): 65-74, 1997.
- ElGR91 Ellis, C.A.; Gibbs, S.J.; Rein, G.L.: Groupware – Some Issues and Experiences. Communication of the ACM 34:1, 38 – 58.
- Fros86 Frost, R.: Introduction to Knowledge Base Systems, MacMillan, New York 1986.
- GeNi88 Gnesereth, M.R.; Nilson, N.J.: Logical Foundation of Artificial Intelligence, Morgan-Kaufmann, LosAltos 1988.
- HaCh94 Hammer, M.; Champy, J.: Business Reengineering: Die Radikalkur für das Unternehmen. Campus Verlag, Frankfurt/Main 1994. (in German)
- HoCo97 Horstmann, C.S.; Cornell, G.: Core Java, Volume I – Fundamentals. Sun Microsystems Press, Mountain View, California, 1997
- JaBu96 Jablonski, S.; Bussler, C.: Workflow Management - Concepts, Architecture and Implementation, International Thomson Publishing, Bonn 1996.
- Kimb96 Kimball, R.: The Datawarehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouses, John Willey &Sohns, Inc., New York 1996.
- MüVe00 Mühlbauer, S.; Versteegen, G.: Studie zum Wissensmanagement, IT Research 2000; www.it-research.net/de. (in German)
- NiPi95 Nippa, M.; Picot, A. (Hrsg.): Prozessmanagement und Reengineering: Die Praxis im deutschsprachigen Raum. Campus Verlag, Frankfurt/Main 1995. (in German)
- Nort98 North, K.: Wissensorientierte Unternehmensführung, Gabler, Wiesbaden, 1998. (in German)
- ReMS00 Reimann, P.; Müller, K.; Starkloff, P.: Kognitiv kompatibel? – Wissensmanagement: Brückenschlag zwischen Technik und Psyche. In: c't Magazin für Computer Technik, 4/2000, S.274ff. (in German)
- Sche98 Scheer, A.-W.: Aris – Business Process Modeling. Springer-Verlag, Berlin et al. 1998.
- Vers00 Versteegen, G.: Was ich nicht weiß ... - Wettbewerbsfaktor: Der richtige Einsatz von Wissen. In: iX – Magazin für professionelle Informationstechnik, 7/2000, S.100ff. (in German)