Multi-Perspective Enterprise Models as a Conceptual Foundation for Knowledge Management

Ulrich Frank

Universität Koblenz-Landau
Rheinau 1, D-56075 Koblenz, Germany
E-mail: ulrich.frank@uni-koblenz.de

Abstract

While successful knowledge management depends on numerous organizational and psychological aspects, the effective documentation, dissemination and utilization of knowledge recommends the introduction of computerized systems to manage knowledge. The design of such a system requires a notion of knowledge that allows to distinguish it from information as it is handled by traditional information systems. In this paper, a pragmatic notion of knowledge is suggested. On the one hand, it is inspired by some characteristics of knowledge stressed in philosophy. On the other hand, it reflects ideas about knowledge as a corporate asset and as subject of organizational learning. Against this background, a number of requirements which should be fulfilled by a system that manages knowledge are developed. They result in suggestions for the content as well as for the architecture of a Knowledge Management System (KMS). Different from organizational memory systems, the proposed KMS features a high level of formal semantics. Different from expert systems or decision support systems, a KMS does not only help with individual problem solving. In addition to that, it provides a medium to foster discourses between people with different perspectives. Both, content and architecture, are inspired by languages that are part of a method for enterprise modelling. To give an idea of the content a KMS provides, some of its various views are illustrated from a user's perspective. After that, the object-oriented software architecture is described in more detail by various excerpts from object models on different levels of abstraction. It emphasizes the reuse of existing, state of the art knowledge and allows for individual revisions and enhancements as well. The architecture also includes an interface level layer that helps with the semantic integration of KMS and traditional IS. Both, content and architecture of the suggested KMS are one out of many possible solutions. For this reason, we will also briefly discuss the pivotal challenges that have to be faced by research in knowledge management if it includes the design of specialized software systems.

1. Introduction

Knowledge management aims at concepts and theories that help to generate, organize and leverage knowledge in social settings. That recommends to focus research in general on cognitive, social, cultural and organizational aspects. In particular, this includes processes of "organizational learning" ([6], [19], [27]), organizational measures to promote the exchange of knowledge (for instance: [20]), incentives for the acquisition, documentation and dissemination of knowledge as well as the reconstruction of specific knowledge domains through "ontologies" [13]. In addition to social and psychological aspects, information technology - if applied in a sensitive way - can be a very effective driver of successful knowledge management. The management of knowledge by specialized software improves availability, offers measures to adapt the mode of access to knowledge and its presentation to individual preferences and allows to deploy specialized software that operates on digitized knowledge. To give a few examples for functions that could be performed by software: retrieval, decision support, simulation, teaching.

Despite the obvious relevance of information technology for knowledge management, there has been only little work on the design of software for knowledge management. The term "organizational memory" has been introduced for a category of information systems that serve to represent knowledge that is subject and result of organizational learning. However, usually the descriptions of organizational memory systems remain on a vague level or they are essentially characterized as hypertext or hypermedia systems (for instance: [7]). On the other hand, there are numerous systems that deal with the representation and application of knowledge. To name a few: decision support systems (DSS), management information systems, expert systems (XPS). Assuming that it makes sense to define a type of software that we could call knowledge management system (KMS), we have to face a number of questions:

- What are the specific characteristics of knowledge compared to information or data?
- What is the difference between information systems in general and KMS?

- What is the difference between KMS and specialised systems like organisational memory systems, knowledge based systems etc.?
- Are there any general requirements, a KMS should fulfil?
- Is it possible to provide a KMS with a generic body of knowledge that can be used in (almost) any company?

The answers that are developed in this paper suggest that there are substantial similarities between knowledge management systems and conceptual enterprise models. Against this background, we will propose an architecture of a multi-perspective KMS as well as its generic content that can be applied to and refined for a wide range of companies. Both, content and architecture, are inspired by a method for enterprise modelling that has been developed during a period of several years ([8], [9]). The architecture is illustrated in more detail by various excerpts from object models on different levels of abstraction. It emphasizes the reuse of existing, state of the art knowledge and allows for individual revisions and enhancements as well. In the long run, it is not desirable to regard a KMS as an isolated system. Therefore, we will outline how to integrate KMS with existing information systems. We will also briefly discuss the chances and pitfalls to be expected from KMSs that become a vital part of corporate knowledge management - in terms of knowledge creation and validation.

2. Subject and Representation of Knowledge

Any attempt to define the term knowledge has to face a dilemma. On the one hand, knowledge - both as part of colloquial and scientific language - seems to be a self evident term with no need for further explanation. Nevertheless you can find many deviating definitions of knowledge. That makes it probably impossible to find a definition that is compatible with most existing notions of knowledge. On the other hand, knowledge represents a phenomenon that is very difficult to reflect upon: While we can speak about knowledge, any insight to knowledge can be regarded as knowledge itself. Similar to language we can differentiate between knowledge and knowledge about knowledge (meta knowledge). But although we can do that over many levels, in the end we cannot avoid a *regressum ad infinitum*. For these reasons, it seems to be a frustrating endeavour to develop a comprehensive definition of knowledge. Fortunately, such a definition is not necessary for our purpose. We mainly need a pragmatic image of knowledge that is suited to differentiate KMS from traditional information systems.

At the same time, it is not satisfactory to simply refer to the colloquial meaning of knowledge. In this case, proposing knowledge management systems would mean to introduce just another label that benefits from the mystification of an impressive term. However, because of the deep complexity of the term, we will not suggest yet another definition of knowledge. Instead we will focus on aspects of knowledge that are relevant for its management by machines and that are suited to deduct essential features of systems we could call knowledge management systems.

2.1 Knowledge and Information: A Preliminary, Pragmatic Differentiation

In philosophy, knowledge is essentially related to cognition, intellectual discovery, explanation and understanding. It is differentiated from beliefs. Therefore there is emphasis on methods to structure and evaluate scientific knowledge ([21], [14], [18], [25]). In sociology, the focus is on the social construction of knowledge, which includes the relationship of knowledge, language and power ([3]). While there is no clear difference to the term "information", the analysis of information is usually stressing other levels of abstraction. Within the engineering disciplines, information theory is focusing on mathematical descriptions of processing and transmitting information. The term information is usually defined as a probability for events ([24]). Semantic information theories (like [2]) on the other hand focus on the formal representation of meaning by symbols. In both cases, information is regarded as formal patterns only.

While these philosophical considerations do not allow to directly distinguish KMS from other types of information systems, they provide some hints: Knowledge is related to describing, analyzing, understanding and eventually changing the world that surrounds us. Applied to the context of this paper - corporate knowledge - we do not consider any real world domain, but only business firms and their relevant surroundings. In addition to that, knowledge in this context is regarded as an organizational and not just an individual asset. That implies adequate measures to communicate and evaluate knowledge. Against this background, corporate knowledge - in contrast (not necessarily opposed) to information - can be characterized by the following features:

Understanding and Reflection

Knowledge implies perception and thinking. Hence, there is emphasis on reflecting upon ways to organize the business and to accomplish competitive advantage. This is different from the information that is needed on an operational level: For pragmatic reasons it will usually be no subject of further intellectual investigation. In other words: Knowledge describes how we *understand* our surrounding. Therefore, knowledge is usually associated with a higher level of abstraction. It is expressed by propositions about classes or concepts rather than by statements about single instances. Information systems typically contain representations of numerous instances. The conceptualization of instances of the same kind (through classes or types) happens usually outside the boundaries of the system that the user has access to.

Scepticism and Evaluation

Knowledge reflects assumptions about actual or possible domains. These assumptions may be more or less adequate or even plain wrong. Applying such a sceptical view suggests that users of such a system should regard the concepts/statements it represents as revisable - instead of taking them for granted. It also recommends to provide comprehensive reasons for the knowledge presented to the user.

Communication and Dissemination

Within an organisation, communication is a prerequisite to evolve and disseminate knowledge, while knowledge in turn can be regarded as a medium to foster communication. It has to be taken into account that - at least in larger organizations - various terminologies or specialized languages are used to express knowledge. Language barriers often cause friction and misallocation of resources. Bridging these gaps requires special knowledge that we could call interface or translation knowledge.

Subject and Result of Organizational Learning

In social settings, individual judgement of situations or concepts is often not sufficient. Instead, it is required that many people agree on common interpretations. Hence, the content of a knowledge management system should be regarded as subject of as well as a medium for organizational learning.

2.2 Some Remarks on Formalisation

Knowledge that can be communicated can usually be represented on a digital computer by deploying adequate types of multimedia representations. However, when it comes to the design of efficient software, representing knowledge merely as sets of symbols is hardly satisfactory: Only if symbols are supplemented by formal semantics, they allow for useful interpretation by a machine. The more semantics is specified for a particular representation, the better are the chances for the implementation of powerful retrieval or navigation mechanisms. Formalization is also a prerequisite for deploying inference procedures, for automatic decision support or for simulation. In addition to that, formalization fosters the integrity of the represented knowledge because it allows to define and control more powerful constraints.

These considerations recommend to formalize knowledge on a high level of semantics. However, formalization does not come without pitfalls. This is especially the case if the knowledge that is stored within a KMS is to express human understanding and reflection about an organization and its surrounding. Within theory of knowledge there has been a long debate about the limits of formalization. Proponents of a hermeneutic approach are convinced that human understanding cannot be completely represented in a formal language. On the other hand, (neo-) positivists argue that in the long run it is possible to reconstruct understanding with substantial explanations expressed in a formal language. While we do not see a chance to establish objective reasons for any of these positions, there is evidence that there are terms/concepts we can talk about (they refer to common knowledge) without being able to formalize their semantics in a satisfactory way. Sometimes, and this is typical for traditional information systems, it is not necessary to formalize the entire semantics of a concept. Instead, an abstraction that deals only with a little part of the concept is sufficient. For instance: The description of a customer or a product is usually restricted to a few (formal) attributes. However, reflecting upon and (re-) organizing a business will usually include concepts that balk at formalization. For instance: competitive advantage, customer satisfaction, motivation, job enrichment etc.

Against this background, it would not be appropriate to restrict the content of KMS to formalized knowledge. Instead, the knowledge represented in a KMS should be formalized as long as formalization is not too expensive or accompanied with dysfunctional bias that is caused by overabstraction/oversimplification. Otherwise it is acceptable to store knowledge with (multi media) representations (natural language descriptions, figures, photos etc.) that do not incorporate formal semantics but that are of value for the human user.

3. Consequences for the Design of Knowledge Management Systems

The features of knowledge we have suggested as relevant for our purpose are abstract in nature. To develop a more concrete notion of a KMS, we will first propose a number of requirements a KMS should fulfil. Against this background, the contents of a KMS and its presentation is illustrated from a user perspective.

3.1 Requirements

In general, a KMS should serve everybody who is involved in processes of understanding, evaluating and (re-) organizing the business. Due to the nature of these tasks, the primary focus groups include consultants (internal and external), new employees who have to understand the company in general and their task in particular, executives, system analysts as well as customers and suppliers that participate in cross-organizational business processes. A KMS should provide these groups with relevant knowledge. At the same time it should support the documentation and exchange of knowledge. The following requirements reflect this purpose in more detail:

Emphasis on Conceptual Level

Rather than providing data about a large amount of instances, a KMS should offer definitions of concepts that are needed for the description and analysis of a corporation. To give a few examples for such concepts: corporate strategy, organizational unit, business process, task, employee etc. Note that these concepts are usually not defined independently from one another. Instead, their semantics will usually include relationships between concepts. For instance: An employee is assigned to one or more organizational units. The concepts have to be accompanied by rules that define how to apply them in order to model certain phenomena - for instance a type of business process. Therefore we could also speak of *languages* and their application.

Reuse of Existing Knowledge

Although there is no unified terminology for the description of corporate knowledge, there are a number of elaborated and well documented concepts available - provided, for instance, by text books of corresponding disciplines. This is also the case for the documentation of relevant causal relationships. A knowledge management system should provide an adequate body of existing knowledge. This is for various reasons. The reuse of knowledge does not only contribute to the economics of a KMS. It should also improve the overall quality of its content. In addition to that it fosters communication by referring to a body of knowledge (we could also say: to languages) many people are familiar with.

Convenient and Safe Adaptation to Individual Needs

Sometimes the concepts provided by a KMS will not satisfy individual requirements. Therefore a KMS should allow to modify concepts. For instance: If the conceptualization of a task within a business process is not satisfactory in a particular case, there should be a way to change its semantics. However, in order to support the integrity of a system, it is not a good idea to allow for arbitrary modifications.

Intuitive Understanding

Often, the knowledge stored in a KMS represents a complex context. Nevertheless, in order to serve its purpose, the system should provide this knowledge in a comprehensive way. The user should be able to understand the contents of a KMS up to a degree he is interested in. For this purpose, a KMS should give the user a chance to reconstruct concepts he does not understand by allowing to trace them back to concepts he is familiar with. An intuitive access to knowledge does not only depend on adequate semantics. The concepts as well as the artefacts (models) the user can design by applying them should also be rendered in a illustrative way. That recommends to offer representations which are common within a target group..

Adequate Level of Formalization

In current practice, graphical representations are often drawings without any domain level semantics. For example: A consultant draws an organizational chart or the model of a business process either by hand or with a graphical tool. For reasons explained in 2.2 such a poor level of semantics is not satisfactory for a KMS. Instead, the knowledge managed within a KMS should be formalized to a degree described in 2.2. That does not necessarily mean, however, that all the details of a formal specification have to be presented to every user.

Support of Multiple Perspectives

In order to support different users and different tasks, a KMS should provide various perspectives on the knowledge it stores. Managing complexity recommends to offer different levels of detail. For instance: Sometimes it will be sufficient to get a description of a business process that is restricted to an outline of the temporal relationships between high level tasks. In other cases it may be important to provide a comprehensive description of every task within the process as well as of the required resources. The plethora of intellectual tasks to be performed in an organization is usually accompanied by a separation of concerns: Classes of problems are related to certain professional communities. In order to support these communities, a KMS should provide concepts that relate to corresponding specialized terminologies and abstractions. To give an example: A system analyst is interested in descriptions of a business process that include data or object models of required and produced information, while a controller may be more interested in concepts that allow to measure the performance of a process.

Integration of Perspectives

While there is certainly need for specialization, it is of crucial importance to foster communication between people with different professional backgrounds. Especially in those cases where there is a cultural chasm between different groups (like business people and information technology professionals), a KMS should help to avoid redundant work and conflicts that block the effective use of resources. This purpose can be served by a conceptual integration of perspectives. Two perspectives are integrated by introducing a set of common concepts. In addition to the conceptual integration through common concepts, the gap between different communities can also be narrowed by providing comments/illustrations of concepts for non-experts. Note that the quest for integration is not restricted to different knowledge perspectives. Knowledge enriches information that is related to it, while information (focusing on the operational level) may contribute to a further illustration of knowledge. That recommends a tight integration with an existing IS.

3.2 Outline of a Multi-Perspective Knowledge Management System

The requirements we have proposed still allow for a wide variety of systems. As with any other IT artefact, it would be presumptuous to claim one best solution. The following outline of a KMS is only a suggestion of how such a system could look like. It is inspired by a method for enterprise modelling called MEMO ("Multi Perspective Enterprise Modelling", [9]). This analogy is not too surprising: Models of an enterprise capture knowledge. At the same time, they should serve as a medium to foster communication between people with different goals, preferences and attitudes. There is only one essential difference: Enterprise models are usually regarded as instruments that are primarily used for the design and the introduction of information systems. In contrast to that we regard a KMS as a system that permanently represents relevant knowledge and that is - at least in the long run - integrated with the corporate information system.

A KMS should represent different perspectives on an enterprise. However, it is not obvious how to identify appropriate perspectives. Publications on enterprise modelling include a number of different suggestions ([26], [22]). In accordance with MEMO we suggest three main perspectives: *strategy*, *organization* and *information system*. Since each of these perspectives is still complex on its own, each of them is differentiated into five aspects: *structure*, *process*, *resources*, *goals*, *environment*.

Fig. 1 illustrates this abstract overview of a multi-perspective KMS. The various foci (a focus is a particular aspect within a certain perspective) within this framework are illustrated by characteristic terms. From a user's point of view, the framework presented in fig. 1 can be regarded as the "main menu" of the system that offers different foci to "zoom" into. One or more foci are represented within a particular view of the KMS. For instance: The foci "Organization/Process", "Organization/Resources" and "Organization/Goals" are represented together within descriptions of business processes. The focus "Information System/ Structure" is represented by an object model. According to our preliminary characterization of knowledge, the emphasis of a KMS would be on a conceptual level. A conceptual level includes descriptions of more or less complex concepts. For instance: a *type* of a business process, a *type* of an organization structure, a *type* of an information structure. In order to foster consistent descriptions, a KMS includes a meta level (we could also say: a knowledge scheme) that provides abstract concepts within specialized modelling languages that are used to describe more concrete concepts. For example: The model of a business process can be created by using a process modelling language. Such a language includes an abstract concept of a business process. It defines that a business process may be aggregated from many other processes. A process may refer to certain resources. One or more organizational units may be in charge of a particular process instance. The meta level should provide generic, state of the art knowledge, comparable to a proven specialized terminology. The conceptual level on the other hand represents knowledge that reflects the specific situation in a particular company. However, the conceptual level allows to reuse existing knowledge from external sources, too. During the last years, the idea of so called reference models for certain types of business firms has gained increasing popularity (for instance: [22]). These models describe perspectives of certain types of business firms - for instance types of organizational structures and business process types for insurance companies. They are to provide carefully developed blueprints to support organizational design and the design of information systems. In practice, reference models are provided with widely spread business software, like SAP/R3. They serve as an additional documentation as well as an orientation for re-organizing the business in order to adapt to the software. Fig. 2 illustrates relevant views, a KMS should provide as well as the relationships between these views. While the views emphasize a graphical representation, they also include textual descriptions.

A multi-perspective KMS does not only provide knowledge on different levels of abstractions and for different tasks. In addition to that, it allows to navigate through an enterprise on different paths. Thereby a user has the chance to enrich his preferred perspective with related ones. Also, it serves as a medium for focused discourses: Starting with the highest level (fig. 2) the different participants of a discourse could literally point to the subject they are interested in. On a more detailed level, the KMS allows to interconnect concepts/instances that are part of different views. Notice that such an approach is different from those organizational memory systems that are implemented as hypermedia systems. A KMS includes more semantics and makes sure that relevant concepts are structured according to common standards (ensured by modelling languages

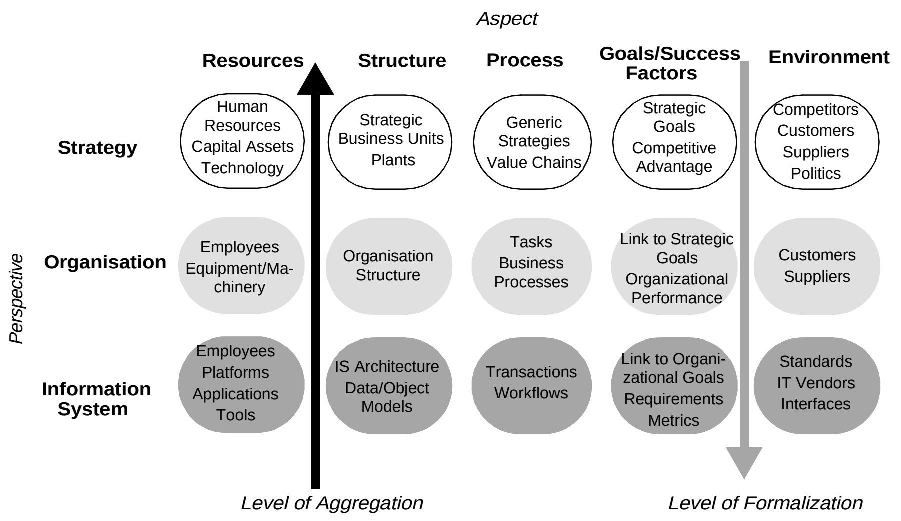

Aspect
Resources
Structure
Process
Goals/Success Factors
Environment
Strategy
Human Resources Capital Assets Technology
Strategic Business Units Plants
Generic Strategies Value Chains
Strategic Goals Competitive Advantage
Competitors Customers Suppliers Politics
Perspective
Organisation
Employees Equipment/Machinery
Organisation Structure
Tasks Business Processes
Link to Strategic Goals Organizational Performance
Customers Suppliers
Information System
Employees Platforms Applications Tools
IS Architecture Data/Object Models
Transactions Workflows
Link to Organizational Goals Requirements Metrics
Standards IT Vendors Interfaces
Level of Aggregation
Level of Formalization

Fig. 1: Apects, Perspectives and related Subjects

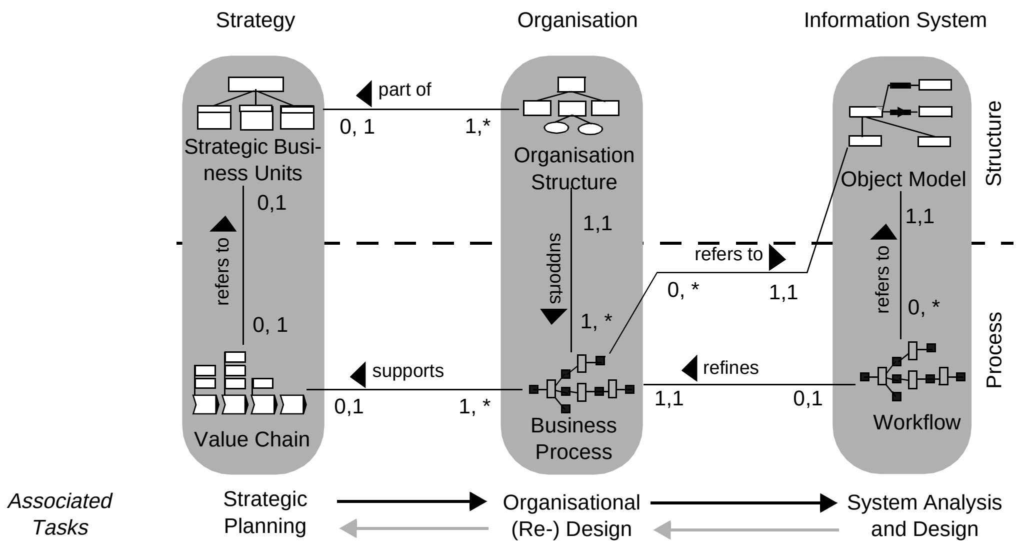
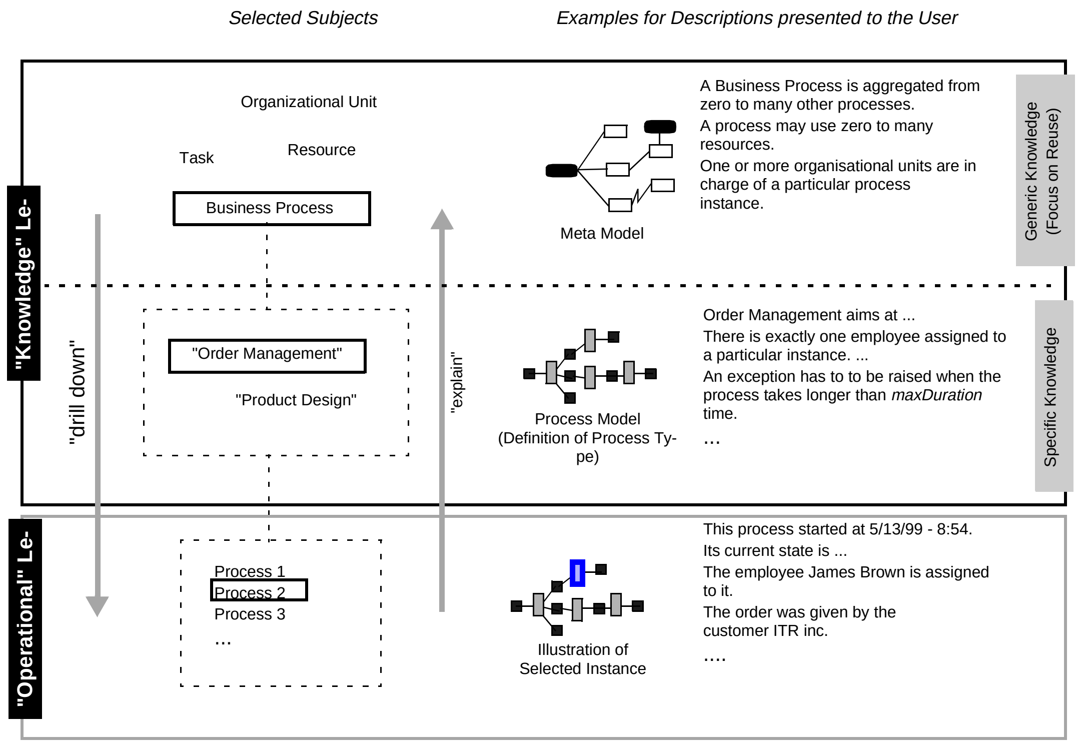


Fig. 2: Views of a KMS and their Relationships



Selected Subjects
Examples for Descriptions presented to the User
Organizational Unit
Task
Resource
Business Process
Meta Model
A Business Process is aggregated from zero to many other processes.
A process may use zero to many resources.
One or more organisational units are in charge of a particular process instance.
Generic Knowledge (Focus on Reuse)
"Knowledge" Le-
"drill down"
"explain"
"Order Management"
"Product Design"
Process Model (Definition of Process Type)
Order Management aims at ...
There is exactly one employee assigned to a particular instance. ...
An exception has to to be raised when the process takes longer than maxDuration time.
...
Specific Knowledge
"Operational" Le-
Process 1
Process 2
Process 3
...
Illustration of Selected Instance
This process started at 5/13/99 - 8:54.
Its current state is ...
The employee James Brown is assigned to it.
The order was given by the customer ITR inc.
....

Fig. 3: Levels of Abstraction within a particular View and Relationship to corresponding Aspects of an IS

and corresponding editors). Therefore it allows for more powerful machine operations. For instance: "show all types of business processes", "show the current corporate strategy", "calculate the bottlenecks within a particular business process type", "show all types of business processes the data processing department is involved in". A user who has selected a particular view may either move to a higher level that provides meta descriptions of the actual level or "drill down" to a lower level. Fig. 3 illustrates the different levels of abstraction provided by a KMS from a user's perspective.

While a KMS clearly stresses these "knowledge" levels, it is important to integrate it with the operational levels stressed by traditional IS, too. From the user's point of view there is an obvious semantic relationship between the conceptual level and the operational level: The operational level manages instances of the concepts defined on the conceptual level. To give an example: On the conceptual level one would describe the business process type "Order Management" (see fig. 3). Within a traditional IS, the user would either explicitly or implicitly deal with a particular process instance. In addition to that, creating and initializing instances could be done in order to provide a more illustrative representation or to prepare for simulation.

4. An Architecture for Knowledge Management Systems

An architecture of a KMS should cover the different levels of abstraction and their interdependencies. The architecture we propose here has been inspired by the specification and use of modelling languages within MEMO. In order to take advantage of the relatively high level of abstraction offered by object-oriented languages, the architecture is designed and specified in an object-oriented way.

The architecture proposed for KMS in this paper consists of three layers. The *meta level layer* serves to specify and manage terminologies or languages as well as the editors used to manipulate models written in these languages. The *conceptual level layer* serves to create and manage models. In other words: It includes instances of the editors as well as instances of the language definitions defined on the meta level layer. In order to allow for a seamless integration with traditional IS, it is necessary to provide an adequate interface. While the concrete definition of such an interface depends on the characteristics of a specific IS, there is a level of abstraction that is typical for IS. The *interface level layer* is to provide concepts on such a level of abstraction.

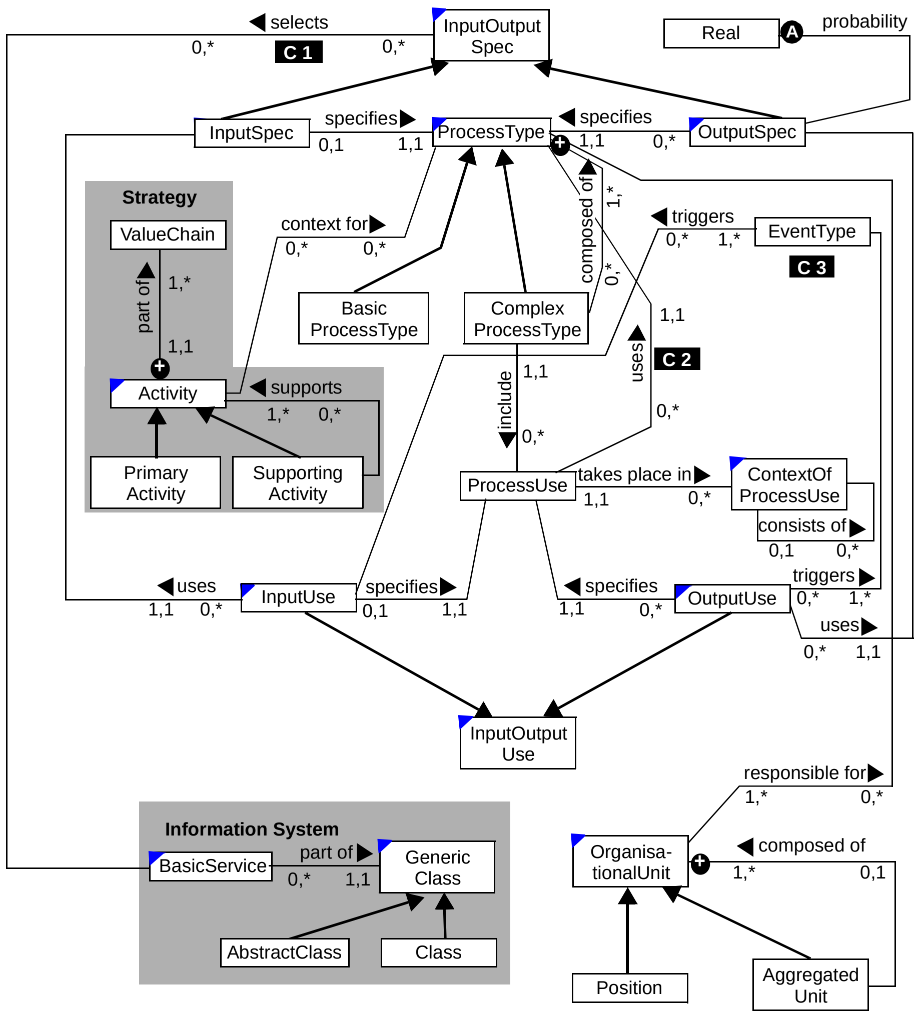
4.1 The Meta Level Layer

This layer, which we could also call the terminology or language layer, consists of classes that serve to reconstruct meta models (language descriptions) and provide additional functionality needed for special purpose modelling editors. At present time, MEMO includes specifications of three special purpose languages: The MEMO Organization Modelling Language (MEMO-OrgML) provides concepts to describe organization structures and associated business processes. The MEMO Strategy Modelling Language (MEMO-SML) is based on the reconstruction of various approaches to guide strategic planning - like Porter's value chain concept (for an overview see [11]). Finally, the MEMO Object Modelling language (MEMO-OML, [Fra98]) is a specialized language for software developers. Hence, it is the language of choice for representing the information system perspective. With respect to the design of a KMS, the MEMO-OML is of outstanding importance, since it is used to reconstruct the other languages for their use within special purpose editors (like an editor to manipulate a value chain or a business process). The object model rendered in fig. 4 illustrates the content of the meta level layer. The object model also shows how the different languages are integrated: by connecting the corresponding object models through common classes. The graphical representation of the object model is accompanied by structured descriptions of classes and formal constraints. Only as an illustration, one constraint in fig. 4 is expressed in the formal language GRAL ([12]). While the meta level can be accessed by every user of a KMS, it should not be modified by users. The languages/terminologies provided by the meta level layer are complex and valuable generic knowledge. The object model in fig. 4 which is only a small excerpt of the overall object model of this layer illustrates the complexity of such concepts. While the extent of this paper does not allow for a comprehensive description of the model, the following outline may give an impression of the meaning associated with the concepts of the meta level.

The key concepts for modelling business processes are represented by the classes ProcessType, ProcessUse, ContextOfProcessUse, InputSpec, OutputSpec and Event. ProcessType is an abstract class that is specialized into two concrete classes: ComplexProcessType and BasicProcessType - with ComplexProcessType being composed of n ProcessType (see the recursive definition in fig. 4). In order to differentiate between many occurrences of the same ProcessType within a ComplexProcessType, we introduced the class ProcessUse. A ComplexProcessType may be composed of many ProcessUse, each of which is assigned exactly one ProcessType. In case the decomposition hierarchy of a ComplexProcessType contains more than one occurrence of a particular ComplexProcessType, there is need to differentiate between the associated ProcessUse. For instance: A business process is composed of n ProcessType "Write User Documentation" which is aggregated from - among other things - the ProcessType "Create Figures". The different occurrences of "Create Figures" within "Write User Documentation" could be differentiated by their associated ProcessUse. To differentiate between identical ProcessUse within different occurrences of "Write User Documentation", every corresponding ProcessUse would be assigned to exactly one ContextOfProcessUse. A ProcessType may require an InputSpec and may produce one or more OutputSpec, which are specifications of the required input (resources, objects, media ...) and the produced output. Other classes on this level serve to model corporate strategies (like ValueChain, Activity ...).

4.2 Conceptual Level Layer

This layers serves to create, edit and store conceptual models. For this purpose, objects are instantiated from the classes that constitute the meta level layer. The editors have to ensure that the models are compliant with the lan-



C 1 Only InputSpecs that are assigned to BasicProcessType(s) can select BasicService(s).

$$\forall cpt{:}V_{ComplexProcessType} \bullet (cpt \rightarrow_{specifiedby \bullet InputSpec} \rightarrow_{selects}) = \varnothing$$

C 2 No cyclic decomposition of ProcessType(s)

C 3 An Event must not be isolated.

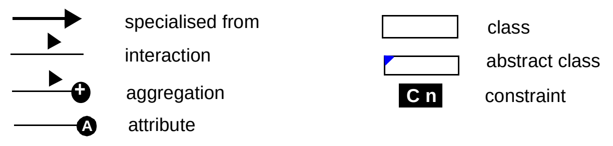


Fig. 4: Excerpt from the Object Model on the Meta Level. The Focus is on the organizational Perspective

guage definitions on the meta level. While the conceptual models are rendered with specialized graphical notations, their semantics is defined within the objects that are instantiated from the meta level classes. Note that these instances (like an instance of BasicProcessType) still describe types, not concrete instances. From a software engineering point of view, this is an important restriction, since it is not possible to derive concrete instances simply by instantiation: Usually, an instance cannot be instantiated from another instance. Different from the meta level, the content of the conceptual level will usually be created and manipulated by the users of a KMS. For this purpose, you would use the various editors to create objects of particular classes and change their state by applying the functions provided by the editor.

This layer requires an adequate runtime system for object management - like a programming language environment, an object-oriented database management system or both. Among other things, object management includes the instantiation and release of objects.

4.3 The Interface Level Layer

Usually, object-oriented languages distinguish between classes and objects. Apparently these two levels are not enough for our purpose: If a concept on the meta level (like ComplexProcessType) is specified as a class, it can be instantiated into a particular object with the name "Order Management". However, it is not possible to create an instance from an instance - that you would need in order to represent a particular process. Within a traditional IS - no matter whether it includes an explicit or rather an implicit notion of a business process, the user would deal with a particular instance that has a specific state. Since the concepts of the interface level layer cannot be created from the conceptual level through instantiation, we suggest to *generate* representations that are appropriate to serve as an interface to traditional IS. We assume that such an IS is specified in an object-oriented way, too. Therefore, the interface level would consist of object models or class definitions, a particular IS could be instantiated from. In addition to static object models, one would also need functional and dynamic descriptions - such as message flow diagrams, state charts or specific languages to describe workflows. With respect to the protection of investment, it is a good idea to use standardized representations, like the Unified Modelling Language (UML, [4]), the Object Definition Language (ODL) as part of the ODMG standard ([CaBa97]) for object-oriented database management systems or the WPDL as part of the WFMC standard [28] for the specification of workflows. Concepts appropriate for the instance level layer can be generated from corresponding concepts on the conceptual level layer. Usually, generation is accompanied by the loss of semantics (certain aspects of the source have to be neglected; this is especially the case with the current version of the WPDL) as well as by adding further semantics (like the specification of user interfaces or information required by a compiler). Unfortunately, that compromises the promises of integration: It is not possible to simply generate a new version of the interface level layer whenever the conceptual level layer has changed. Therefore generation recommends to apply restrictive policies for code or model management. In order to overcome this problem, it is necessary that standardized languages which are used to specify schemata for IS provide richer semantic concepts.

Usually, an IS will have been developed before a KMS. In this case, the (hopefully) existing schema information of the IS has to be mapped to the generated schema (which will usually imply a further loss of semantics).

5. Conclusions

Based on a pragmatic notion of knowledge - which, nevertheless, is in line with some characteristics of knowledge stressed in philosophy - we have proposed the concept of a multi-perspective KMS. It is characterized by an object-oriented architecture as well as a generic body of knowledge. This is different with other systems that deal with corporate knowledge. DSS are not associated with a specific architecture. Often, publications about DSS do not even touch this topic. XPS usually feature a specific architecture that is based on the separation of a declarative knowledge base and an inference engine. In most cases, they are not based on an object-oriented architecture. Therefore the integration with an existing IS is often hard to accomplish. Different from DSS and XPS, a multi-perspective KMS is not only directed at certain groups of users, like executives, but at a wide range of people with different skills and different needs for supporting knowledge.

In the long run, it is desirable to regard a KMS as an integral part of a corporate IS. Such an integration promises a number of advantages. New users of an IS as well as new employees in general can access the "knowledge level" to get a deeper understanding of the corporation. In this respect, a KMS is similar to the concept of "dual information systems" [16] that are intended to enable users to create and share knowledge with others. Since a KMS offers different perspectives on an enterprise which are interrelated through common concepts, it provides a medium to foster discourses between people with different perspectives. In general, this helps to promote processes of organizational learning - in the sense that people are supported to understand perspectives of people with a different professional background. In particular, a KMS can also contribute to overcome the common barrier between business people and information technology professionals that Keen regards as "the one factor that can block the effective use of computers and communications." [17]

Sometimes it is argued that one priority goal of research in knowledge management should be to make "tacit knowledge visible and concrete" ([15]). However, from our point of view, such a statement is misleading for at least two reasons. Firstly, it has been the central objective of any scientific research to enhance the amount of visible and concrete knowledge - no matter whether is was "tacit" before or not. Secondly, the pivotal challenge, knowledge management in general, the successful introduction of knowledge management systems in particular, has to face is different anyway. Similar to methods for enterprise modelling, the success of a KMS depends essentially on the acceptance of the languages that it provides as well as on the acceptance of the models created with these languages. At present time, there is a vast amount of - more or less precisely specified - graphical languages that are used to render certain aspects of a company. However, this variety is hardly necessary. Instead, it is - at least to a great extent - the result of arbitrary decisions. To allow for a higher level of knowledge reuse and a better integration of different KMS, research has to focus on terminologies/languages - or so called "ontologies" - which are appropriate for most companies. This assumption constitutes a chance and a challenge at the same time. It is a chance, because we do not have to find the one best way to conceptualize a business process or a corporate strategy. It is a challenge, because it is certainly not sufficient to suggest just any language. The idea of scientific progress implies that there are criteria that allow to discriminate between competing results of research. The evaluation of a language, however, is a delicate task: It requires the use of this language which in turn will bias our perception and judgement.

References

[1] Balzer, W.; Moulines, C. U.; Sneed, J. D.: *An Architectonic for Science: The Structuralist Program*. Reidel, Dordrecht, 1987

[2] Bar-Hillel, Y. (Ed.): *Language and Information: Selected Essays on their Theory and Application*. Addison-Wesley, Reading, Mass., 1964

[3] Berger, P. L.; Luckmann, T.: *The Social Construction of Reality: A Treatise in the Sociology of Knowledge*. Doubleday, New York 1967

[4] Booch, G.; Jacobson, I.; Rumbaugh, J.: *The Unified Modeling Language User Guide*. Addison-Wesley, Reading, Mass., et al., 1998

[5] Cattell, R.; Barry, D. K.; Bartels, D. (Eds.): *The Object Database Standard: ODMG 2.0*. Morgan Kaufmann, San Francisco, Ca., 1997

[6] Easterby-Smith, M.: "Disciplines of organizational Learning: Contributions and Critiques", *Human Relations*, Vol. 50, No. 9, 1997, pp. 1085-1114

[7] Euzenat, J.: "Corporate Memory Through Cooperative Creation of Knowledge Bases and Hyperdocuments", *Proceedings 10th Banff Workshop on Knowledge Acquisition for Knowledge-Based Systems*. SDRG Publications, Calgary, 1996, pp. 1-18

[8] Frank, U.: MEMO: "A Tool Supported Methodology for Analyzing and (Re-) Designing Business Information Systems", Ege, R.; Singh, M.; Meyer, B. (Eds.): *Technology of Object-Oriented Languages and Systems*. Prentice Hall, Englewood Cliffs, 1994, pp. 367-380

[9] Frank, U.: *Enriching Object-Oriented Methods with Domain Specific Knowledge: Outline of a Method for Enterprise Modelling*. Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 4, Universität Koblenz 1997

[10] Frank, U.: *The MEMO Object Modelling Language (MEMO-OML)*. Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 10, Universität Koblenz 1998

[11] Frank, U.: MEMO: *Visual Languages for Enterprise Modelling*, Nr. 17, Universität Koblenz 1999

[12] Franzke, A.: *GRAL 2.0: A Reference Manual*. Fachberichte Informatik. Universität Koblenz 1997

[13] Gruber, T.: *Toward Principles for the Design of Ontologies used for Knowledge Sharing*. Stanford Knowledge Systems Laboratory Report KSL-93-04, 1993

[14] Habermas, J.: T*he Theory of Communicative Action: Vol. 1: Reason and the Rationalization of Society*. Beacon Press, Boston, Mass., 1985

[15] Holtshouse, D.: "Knowledge Research Issues", *California Management Review*, Vol. 40, No. 3, 1998, pp. 277-280

[16] Käkölä, T.K.; Koota, K.I.: "Redesigning Computer-Supported Work Processes with Dual Information Systems: The Work Process Benchmarking Service". Journal of Management Information Systems, Vol. 16, No. 1, Summer 1999, pp. 87-119

[17] Keen, P.W.: *Shaping the Future. Business Design through Information Technology*. Harvard Business School Press, Boston, Mass., 1991

[18] Lakatos, I.; Musgrave, A. E. (Eds.): *Criticism and the Growth of Knowledge*. Cambridge University Press, Cambridge, 1970

[19] Nicolini, D.; Meznar, M. B.: "The Social Construction of Organizational Learning: Conceptual and Practical Issues in the Field", *Human Relations*, Vol. 48, No. 7, 1995, pp. 727-746

[20] Nonaka, I.; Konno, N.: "The Concept of 'Ba': Building a Foundation for Knowledge Creation", *California Management Review*, Vol. 40, No. 3, 1998, pp. 40-54

[21] Popper, K. R.: *Conjectures and Refutations: The Growth of Scientific Knowledge*. Basic Books, New York, 1962

[22] Scheer, A.: *Business Process Engineering*. Springer, Berlin, Heidelberg, New York, et al., 1994

[23] Schein, E. H.: "Three Cultures of Management: The Key to Organizational Learning", *Sloan Management Review*, Vol. 38, No. 1, 1996, pp. 9-20

[24] Shannon, C. E.; Weaver, W.: *The Mathematical Theory of Communication*. University of Illinois Press, 1962

[25] Sneed, J. D.: *The Logical Structure of Mathematical Physics*. Kluwer Academic Publishers, Dordrecht, 1979

[26] Sowa, J.; Zachman, J.: "Extending and formalizing the framework for information systems architecture", *IBM Systems Journal*, Vol. 31, No. 3, 1992, pp. 590-616

[27] Tsang, E. W.: "Organizational Learning and the Learning Organization: A Dichotomy between descriptive and prescriptive Rresearch", In: *Human Relations*, Vol. 50, No. 9, 1997, pp. 73-90

[28] WfMC (Workgroup 1): *Interface 1: Process Definition Interchange WfMC TC-1016*, Version 1.0 Beta, May 29, 1996. (http://www.aiai.ed.ac.uk/WfMC/ 1996)