An Enterprise Reference Scheme for Integrating Model Based Knowledge Engineering and Enterprise Modelling[1]

Stefan Decker[a], Manfred Daniel[b], Michael Erdmann[a], and Rudi Studer[a]

[a]Institute AIFB, University of Karlsruhe (TH), D-76128 Karlsruhe
e-mail: {decker | erdmann | studer}@aifb.uni-karlsruhe.de

[b]Fachhochschule Schmalkalden, D-98574 Schmalkalden
e-mail: daniel@informatik.fh-schmalkalden.de

Abstract

In recent years the demand on business process modelling (BPM) became apparent in many different communities. To provide a unifying framework for different needs on enterprise modelling we define an enterprise reference scheme and show how the development of knowledge based systems can be incorporated in such a framework. From this framework conclusions for tool support are drawn.

1 Introduction

In recent years the demand on business process modelling (BPM) became apparent in many different communities, e.g. information systems engineering [Sch94], requirements engineering [KiB94], software engineering and knowledge engineering (e.g. [SWH+94]). This suggests to aim at a unifying view on business process modelling in all these disciplines. To achieve the business goals some problems which obstruct these goals must be solved. This can be done either by restructuring the business process, by application of standard software, or by developing individual software components, such as knowledge based systems (KBSs). To be able to model business goals and to analyse problems occurring during the business processes these processes including organisational structures and activities have to be modelled. This is also true when building a KBS in an enterprise environment. Because the KBS is only a small part of the whole business organisation, it must be embedded into or at least linked to all relevant business processes, i.e. it should not be a stand-alone solution. For this purpose we extend the MIKE approach [AFS96] in the BMBF project WORKS (**W**ork **Or**iented Design of **K**nowledge **S**ystems) by offering business models for modelling relevant aspects of an enterprise. Although there are many approaches for enterprise reference schemes (e.g.[HBM+96][KiB94][RaV95][Sch94]), none of them seems completely appropriate for our purposes: most approaches dot not consider KBS as a

1. The work reported in this paper is partially supported by the BMBF (German Ministry of Education, Science, Research and Technology) under grant number 01HS014.

possibility for improving an enterprise's processes (for a general framework, see e.g. [DES96]) or the enterprise model is not elaborated enough. To be able to define an integrated framework including other possibilities to improve business processes (e.g. development of information systems) we determine the standard views of an enterprise and suggest a common notation for all views. This works extends the MIKE approach to cover organisational aspects. To reach this goal, we define relevant views of an enterprise and define the integration of the enterprise views with the models of MIKE. The ideas are illustrated by an example, derived from several elicitations.

1.1 The MIKE Approach

MIKE (Model-based and Incremental Knowledge Engineering) ([AFS96]) defines an engineering framework for eliciting, interpreting, formalizing, and implementing knowledge in order to build KBSs. It aims at integrating the advantages of life cycle models, prototyping, and formal specification techniques into a coherent framework for the knowledge engineering process. Subsequently, we will discuss the main principles and methods of MIKE.

In contrast to other approaches which assume that the expert creates the model himself, it is assumed that the knowledge engineer is the moderator of this modelling process. Considering knowledge engineering as a modelling activity implies that this process is *cyclic*, *faulty* and *approximative*.

Within the modelling process a large gap has to be bridged between informal descriptions of the expertise which have been gained from the expert using knowledge elicitation methods and the final realization of the KBS. Dividing this gap into smaller ones reduces the complexity of the whole modelling process because in every step different aspects may be considered independently from other aspects.

The knowledge gained from the expert in the elicitation phase is described in natural language. It mainly consists of interview protocols, protocols of verbal reports, etc. These knowledge protocols define the *elicitation model* ([Neu93]). This knowledge represented in natural language must be interpreted and structured. The result of this step is described semi-formally in the so-called *structure model* ([Neu93][Neu94]), using predefined types of nodes and links. The structure model consists of four contexts: the concept context defines the domain terminology, the activity context defines the task decomposition, the data flow context defines the data flow between the subtasks and the ordering context defines their control flow.

According to the KADS approach the knowledge-level description of the functionality of the system is given in the *model of expertise* (cf. [SWB93]). For describing the model of expertise in a formal way the formal and operational specification language KARL ([Ang93][Fen95][FAS97]) has been developed. KARL is based on first order logic and dynamic logic and offers language primitives for each of the three different layers of the model of expertise. The contexts of the structure model correspond to the domain-, task-, and inference layer of KADS' model of expertise.

The model of expertise finally includes all functional requirements of the desired system. For the realization of the final system, additional requirements have to be considered which are still independent of the final implementation of the system. These requirements are non-functional requirements such as efficiency of the problem-solving method, maintainability of the system, persistency of data etc. ([LaS95]). Capturing such decisions within the *design model* divides the gap between the model of expertise and the implementation of the final system. For the description of the design model the language KARL has been extended to the language DesignKARL [Lan94] which allows to describe data structures, algorithms and offers additional structuring primitives like clusters and modules.

2 Enterprise Modelling

2.1 Notation

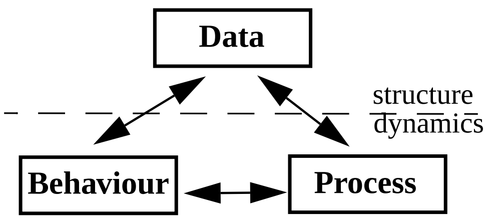

Fig. 1 . Model Perspectives

It is generally accepted that for an operational description of a system three views are sufficient (see figure 1 taken from [RaV95]). These three perspectives have a more principal relationship to modelling: they are generally used to describe the kind of the modelled information (structure vs. dynamic), and therefore can not be used to identify useful views of an enterprise. For example dynamics can be identified in several parts of an enterprise and therefore also in several views (e.g. in the business processes and in the processes, that are executed in a software system). Although the level of abstraction is different in these two processes and they are probably modelled in different layers of an enterprise model, the same notation can be used for both. A notation for modelling an enterprise should fulfill the following objectives: it should be understandable and widely accepted, it should be useful for different types of software systems (e.g. information systems and knowledge based systems) and powerful enough to model all relevant aspects. At last it should bridge the gap between the user world and the developers world. OMT (Object Modelling Technique) (cf. [RBP+91]) has proved its usefulness in several areas: software system design, design of knowledge-based systems [ScW93a] and enterprise modelling ([BKM94], [KKM95]). For these reasons we use OMT in our integration approach. The data constituent in figure 1 corresponds to the static object model of OMT, the behaviour constituent corresponds to the dynamic model and the process constituent corresponds to the functional model. So statecharts are used for the behaviour constituent and DFDs (dataflow diagrams) are used for the process constituent.

2.2 Views

2.2.1 Introduction to the Views

Models mostly have the objective to simplify complex realities by representing only aspects relevant for decisions or actions. Depending on the ensemble of aspects or objects of the reality which are observed, different views reflected by a model are distinguished. In the WORKS approach, nine different views are introduced. The selection and definition is on the one hand determined by the example of well-known views (e.g. in business administration) and on the other hand by the special aims and questions considered in WORKS.

For organisation modelling, the distinction between *organisation structure* and *organisation processes* is useful. In addition, for a work examination, people working in the organisation (*staff view*) and their working tools (*working tool view*) are relevant. The *data view* is a standard view of organisations, when the development of information systems is concerned. So to speak, data are working objects of information processing activities. Communication and cooperation play a special role under criteria of task design (e.g. task splitting between human and computer [Dan93]) and are therefore treated as a particular view (*communication and cooperation view*).

The *expertise view* is founded on the special focus of the WORKS approach on knowledge-based systems. It is the adoption of a standard modelling concept (MIKE or CommonKADS [SWH+94]). In connection with this, the usable knowledge sources (*source view*) are interesting for the purpose of knowledge acquisition. Transverse to the former views, so to say, the *strong points'-/deficient points' view* is located, in which valuing statements concerning organisational, technical and work engineering matters are collected ([DuV92], [DaF92]) . The last view is not further considered in this paper, because its contents is largely informal (e.g. natural text). In the next sections some of the views are introduced in more detail.

2.2.2 The Data View

The data view (figure 7 part b) is essentially a model of the static object model of OMT: Data can be grouped by relations. Special relations are specialisation, aggregation and associations. Both can be enriched by attributes. In applications a much more enriched reference scheme may be required, but it is straightforward to construct one out of these modelling primitives.

2.2.3 The Process View

The process view (see figure 7 part a) is the heart of an enterprise model: it describes the dynamic aspects of an organisation with two main constituents: "process" and "task". The overall modelling activity starts by modelling the business goals. These business goals have to be achieved by business processes [SWH+94]: a process results in a task decomposition of the original goal and the definition of the control flow and the data flow between the subtasks. The subtasks (defined by the business process) can either be business tasks or job tasks etc. We define three levels of processes, which we want to distinguish for enterprise modelling: the business level, the job level, and the job part level, but each of these levels can have an arbitrary depth (on the other hand the process view of an enterprise should not be too detailed, but focus on relevant aspects). The business level deals with processes related to several departments, the job level with rather small processes usually done in one department, and the job part level with processes done by one person to achieve a particular goal. To achieve an integrated modelling technique we use the dynamic and the functional model notation of OMT to describe the data flow and the control flow of a process, i.e. DFDs (dataflow diagrams) and statecharts. By this we adopted the approach of [BKM94]: by an unconventional use of notation the task decomposition is modelled with the object view of OMT. The modelled "subtask classes" are used as processes in the dynamic model description (the control flow).

2.2.4 Organisational Structure View

The organisational structure view is intended to capture the static organisational aspects of an enterprise. The first thing to model is the structure of the organisational units. Therefore we model a decomposition of the organisational unit class into smaller organisational units. Another relation exists between organisational units and jobs or collection of jobs. A job is also related to a collection of tasks, which is a connection from the organisational structure view to the process view. An organisational unit is also related to a set of tasks (modelled in the process view) which it fulfills. Further we want to differentiate between jobs and job places, because both are important to take into account for human needs (and possible problems to solve). To allow statements about larger enterprises the job type and organisation unit type class are used to allow statements about collections of objects.

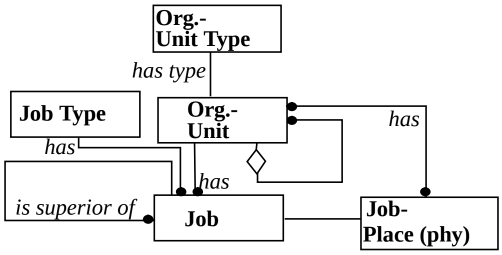

Fig. 2 . Organisational Structure View

2.2.5 Staff View

The staff view allows us to incorporate a human centric view into the modelling process. This covers several aspects: e.g. the engineering of human centric business processes as well as the design of ergonomic software products. For this purpose the two classes "Role" and "Ergonomic Requirements" were defined. Role objects capture the relation between employees and the jobs. In some modelling approaches the "staff view" and the "working tool view" are mixed together into a "resource view". This is not an appropriate choice: modelling should always be done with requirements of the staff in mind. Processes and information systems have to be designed explicitly to conform with human ergonomic requirements.

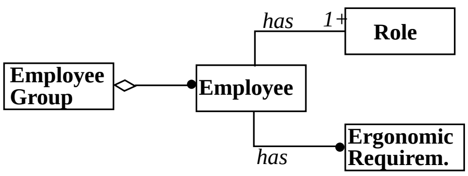

Fig. 3 . Staff View

2.2.6 Working Tool View

The working tool view allows to model dependencies between a task and a tool that is needed to perform it. To design business processes one should know the tools used in the process to detect e.g. media breaks. Therefore we need additional attributes to characterize working tools: type can be e.g. computer (in general), knowledge

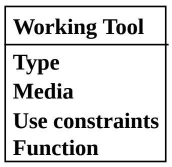

Fig. 4 . Working Tool View

based system (more specific) or a type writer. The attribute "media" says something about the media used in or with that tool, e.g. electronic documents (for a computer system). So there exists a link from the "Working Tool" to the data view. With the attribute "use constraints" restrictions of the working tool can be expressed: e.g. a computer system is not usable during system maintenance. The function attribute should give information about the role of the working tool in the business processes, e.g. give answers to the following questions: Is it essential? For which tasks is the tool usable?

2.2.7 Expertise View

The expertise view (figure 7 part c) is oriented towards the structure model of MIKE and the model of expertise in CommonKADS [SWH+94]: A task is solved by a problem solving method, which needs domain knowledge. The problem solving method defines a task decomposition ("divide and conquer") of the original task and the data - resp. the control flow of these subtasks. The expertise view is a special view: it is the only one, which contains all different model views (cf. figure 1). This is due to the fact, that it represents a complete description of a knowledge based system. So all the other views realize only the frame of the expertise view. The expertise view is of course generally much more complex - however, the integrational approach is very much the same, even in more complicated situations. More advanced techniques for modelling problem solving methods (e.g. ontology mappings [SEG+96]) are not considered in the basic approach.

2.2.8 Source View

The aim of the source view (figure 5) is to provide a possibility to model relevant sources for the knowledge elicitation process. Therefore it is one of the model constituents, that are necessary for the development of a knowledge based system. It supports the planning of the knowledge elicitation process, where e.g. different staff members have to be interviewed. Therefore, we distinguish between sources of the knowledge and the knowledge itself. Sources can be e.g. employees (which are active) but also books, which are passive. After taking the decision, what knowledge is relevant for a business process improvement, it can be determined which knowledge should be elicited and modelled and what sources are relevant for that knowledge. The knowledge can be further classified: Which field does it belong to? Is it directly available and clear (how much effort has the knowledge engineer to put into the elicitation)? The answers to these questions can be modelled using the different attributes.

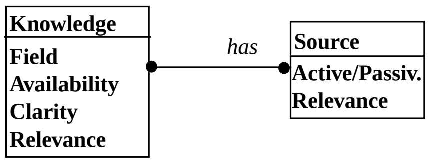

Fig. 5 . Source View

2.2.9 Communication/Cooperation View

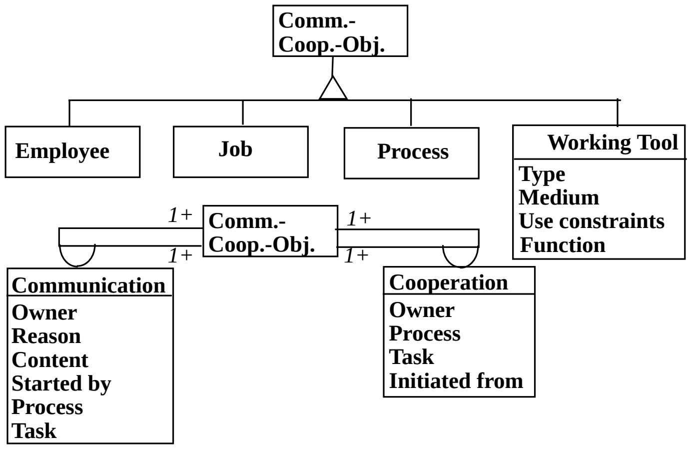

Fig. 6 . Communication/Cooperation View

For the development of an information system as well as for a knowledge based system it is important to know, at which point in the work the employee needs additional information to perform his task. The design of the communication/cooperation view is similar to known techniques of describing human/computer interaction (interaction diagrams) [RBP+91]. The communication/cooperation objects can be instances of the classes employee, job, process, working tool, i.e. these are the objects, that can communicate/cooperate with each other. This provides the possibility of several levels of detail for modelling. The lower part of the diagram defines a relation and corresponds to a simple link between two of these objects: the link is annotated with attributes, which make assertions about the owner, the contents etc. of the communication/cooperation. If two persons communicate/cooperate with each other, they do this usually within a task. To model the information flow, this task/process dependency can be noted in an attribute.

2.2.10 Connections between the views

Several connections exist between these views: most of them are standard connections, but a few are important in the context of the development of knowledge based systems. The most important one is the connection between the process view, the expertise view and the data view (cf. figure 7). The point, where a knowledge based system can support an employee is at the job part task level. At this level an employee works on a closed task, where mainly his knowledge determines how to solve the task. This is the point, where a knowledge based system may come into the game. The tasks performed by problem solving methods are just subtasks of the task the employee perfoms.

Another important link is the connection between the data view and the domain layer in the expertise view: an employee (the expert) does his job in the context of the enterprise, especially in the context of its data. So the input-output of his problem solving behaviour consists mostly of data elements of the data view. Therefore the domain layer of the expertise view has a nonempty intersection with the data view of the enterprise. To perform the knowledge elicitation task the links between the working tool view, the staff view, and the source view are important. These deliver the information, which persons have to be interviewed.

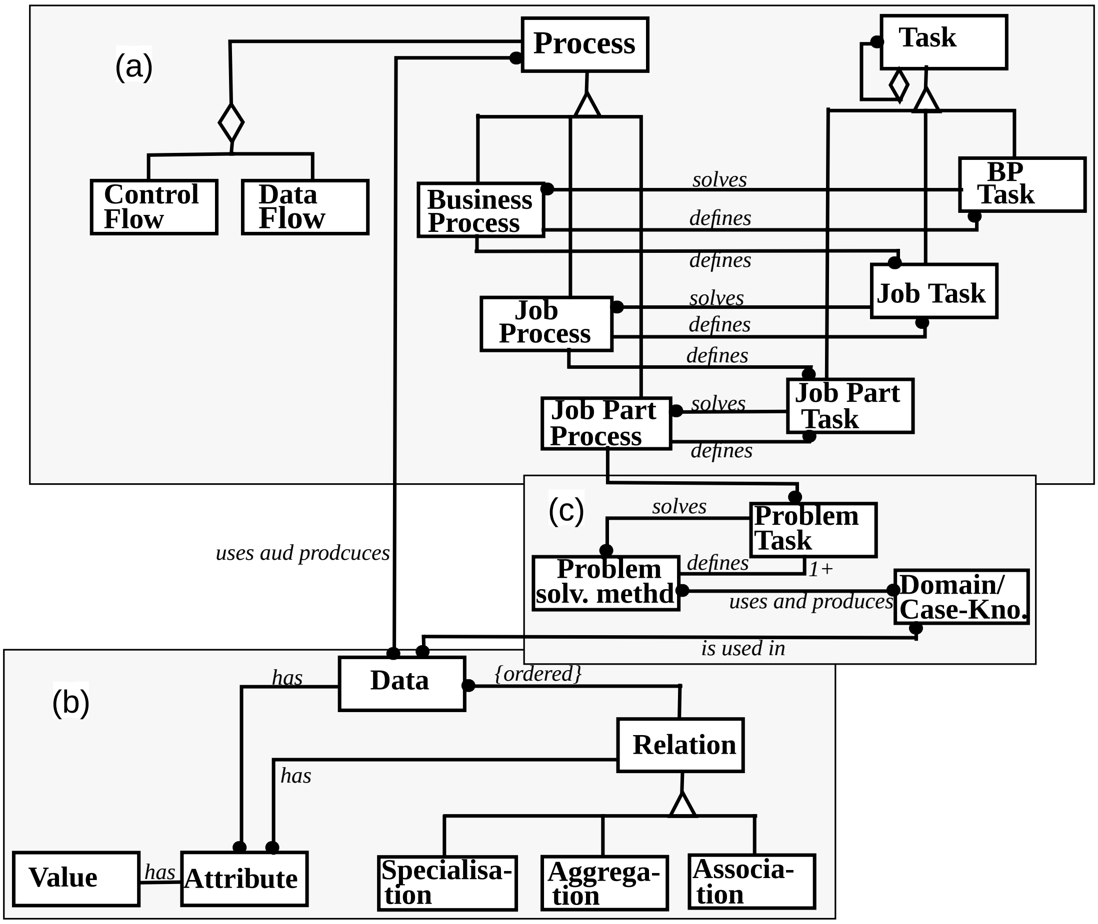

Fig. 7 . Core Relations between (a) Process View, (b) Data View and (c) Expertise View

3 Knowledge Based Systems and Organisation Modelling

Having developed a framework for modelling business processes in general and for embedding them into an organisational environment the question arises which part of a business process could or should be handled by an assisting KBS and not for instance by an information system. Due to the nature of a KBS there does not exist a complete checklist for answering that question. Nevertheless, a few characteristics may be identified: In our framework (the process view) part of a job task is amenable to such an assistance by a KBS. I.e. we do not envision that a complete business process is supported by a KBS. Rather, a task which is handled by a single person or few cooperating persons is a candidate task.

If there exists a completely formal model for specifying the task and for computing a corresponding solution, for instance an optimization model as known from operations research, there is typically no need for a KBS approach. Instead, a KBS approach is advised "when we do not have overt domain and problem solving models" [ShG92].

If domain and task specific problem solving knowledge, which "encodes" the experience of an expert, is needed in order to be able to solve the task in an efficient manner,

such a task is a candidate task for KBS support. "In simple terms this means analysis is not simply interested in what happens, as in conventional systems, but with how and why" [Bro86]. In other words, expertise is concerned with knowing how to do things [ScB96] and is captured in domain and task specific heuristics.
Typically, candidate tasks represent problems which are at least NP-hard in their general formulation [Neb96]. Therefore, experts use their heuristic knowledge for instance to restrict the original problem, to reformulate it or to provide only an approximate solution.
It should be clear that there does not exist a strict borderline between tasks which are suitable for KBS support and those which are not. Therefore, it is up to the business process analyst to make a final decision. Obviously, such a decision will be influenced by a lot of additional aspects, e.g. whether one has already gained experience in developing assisting KBS.

4 Tool Support for Enterprise Modelling

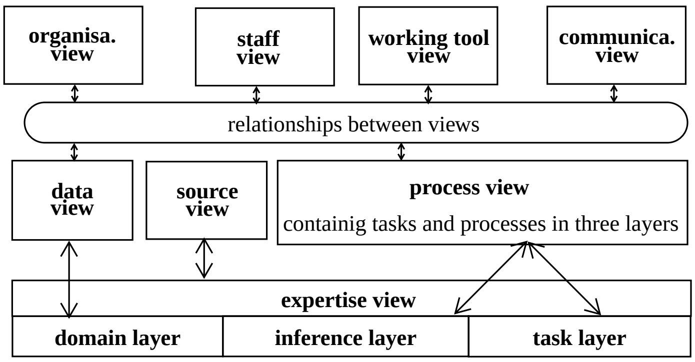

Fig. 8 . Architecture of the Views in a Tool

Our approach stresses the importance of the organisational environment, esp. the primary character of business goals. This organisational environment has to be modelled whether a KBS has to be built, a workflow management system is projected, or the business processes are reorganized in any other way. The construction of the above described views should be supported by an appropriate tool. This claim is realized for example by the ARIS-Toolset [Sch94]. The ARIS model contains slightly different views and concepts and thus the tool set as well. But in principle this tool set can be used to develop an enterprise model which serves as the base for the decision whether to build a KBS or any other means of reorganisation. ARIS is not specifically headed towards building KBSs: it does not support the modelling of the expertise view.
Therefore we extend our MIKE-Tool, which then contains mainly three different sets of views. The first view is the so called elicitation model (not contained in figure 8). The second subset consists of those views which serve to model the environment, i.e the organisational view, the staff view, the working tool view, the communication/ cooperation view, the data view, the source view, and mainly the process view. All these views are interrelated by several relationships (as outlined in figure 8). The third subset consists of the expertise view, which is decomposed into three different layer (domain layer, inference layer, and task layer). The two sets are connected mainly via the process view, the data view, and the source view. So this tool describes business processes and tasks and relates them to problem solving methods and tasks of the expertise view and the data of the enterprise as part of the domain layer of the model of expertise resp. Following MIKE's philosophy of modelling, also for the enterprise

views an elicitation has to take place: the result of this elicitation consists of natural language protocols, images or sound files. The techniques, that MIKE offers for constructing the structure model, can also be used to construct the semiformal enterprise views: The informal protocols are structured and interpreted to constitute the different views. These views are linked to one another by defining relationships between related entities. Furthermore a certain kind of link (*elicitation link* [Neu94]) is established automatically between the protocols and the structured information. Thus everything that is modelled can be traced back to the protocols and thus is put into the correct context. By that inconsistencies and failures during modelling can be found and the communication between the modeller (knowledge engineer) and the information provider (expert) becomes easier.

The business modelling process was started because certain problems arose which obstructed business goals. The areas surrounding these problems and goals should be modelled in more detail than other (possibly less relevant) areas. If a relatively stable state is reached a decision must be made which states how to solve these problems. If the decision yields constructing a KBS then the second subset of views becomes relevant. Further information must be elicited to model problem solving behaviour, so further protocols are produced which complete the input for the expertise view. Now MIKE's structure model is defined. This is done by identifying entities relevant both in the expertise view and in the business views and linking them. Also all elicited protocols may contain relevant information for defining a problem solving process. Largely this process resembles the regular specification process in MIKE, i.e. informal information (from protocols) is interpreted and structured to yield a semi-formal model. The main difference lies in the fact that also semi-formally modelled information contained in "outside" models (i.e. the business views) has to be considered in the structuring process. In that way the higher level business views are closely connected with the structure model in the expertise view. The next step of modelling in MIKE consists or formalizing the semi formal structure model to constitute the formal model of expertise specified in the language KARL. This specification can be tested because KARL is an operational language so that the KBS may be evaluated (in the business setting) by prototyping.

5 An Example Model

5.1 Introduction

To illustrate the ideas we present an example model using the primitives of OMT. The example was derived during the WORKS project. The aim of the project was to build a KBS to support industrial designers concerning ergonomic questions. However, in the course of the project it turned out that "just" a KBS is not enough: large portions of knowledge can only be represented informally, knowledge is built through lessons learned and case studies, and many process steps in the design process have to be supported. Therefore we are aiming now for a knowledge management system, incorporating many techniques. For the requirements elicitation of this system the process of industrial design was modelled and analysed for support needs in some design companies. Parametric design problem solving (see [MoZ96]) was found as one possibility to support the design process. In the following we present a simplified version of these process models and show, how a specific problem solving method (modelled in the expertise view) is integrated into the whole design process. We elicitated several views, but due to the lack of space (and because the modelled companies were rather small), we only present the core views (data, process and expertise view).

5.2 The Data View

During the design process several documents are created or used (cf. figure 9), e.g. a briefing document delivered by the customer specifying initial requirements for the design object, general information about similar products or usable components in the design task, a design object description where e.g. drawings or the component architecture is defined and elaborated. Physical models of the design are created and tested against the requirements. The documents created in the design task have usually a special structure. This structure can be made explicit.

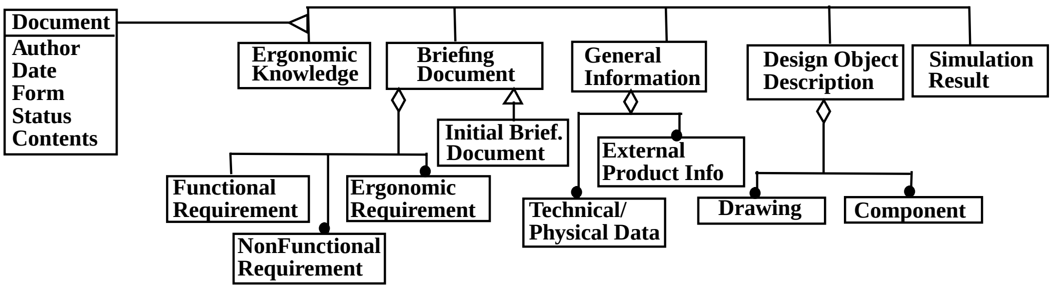

Fig. 9 . Example Data View

5.3 The Process View

The design process modelled in figure 10 is usually performed by one person, so it is located on the job part level (figure 7). The task "design process'' is decomposed into seven subtasks. To elaborate the process model we have to define the control flow (by statecharts) and the dataflow between the subtasks. One subtask is viewed as a state of the overall task "design". Events in the statechart (depicted with the arrows) specify when the task performs a state transition from one state to another (see figure 10 part 2). Dataflow diagrams as used here resemble KADS' inference structures (cf. [ScW93]) and use the data view for defining the concepts of the dataflow.

5.4 The Expertise View

The simplified design process presented above shows several opportunities for computer based support, e.g. document management, organisational memory, CAD, etc. We focus now on a knowledge based system aiming at supporting the "Technical & Functional Elaboration" step of the design process. At this step of the design process most components, possible assignments for the components, constraints and requirements are known. This configuration leads directly to problem solving methods for parametric design ([MoZ96]). We integrate the first decomposition level of the problem solving method "CMR" (complete model then revise) into the design process (cf. figure 11). At this decomposition level the PSM is rather simple. However, it is useful

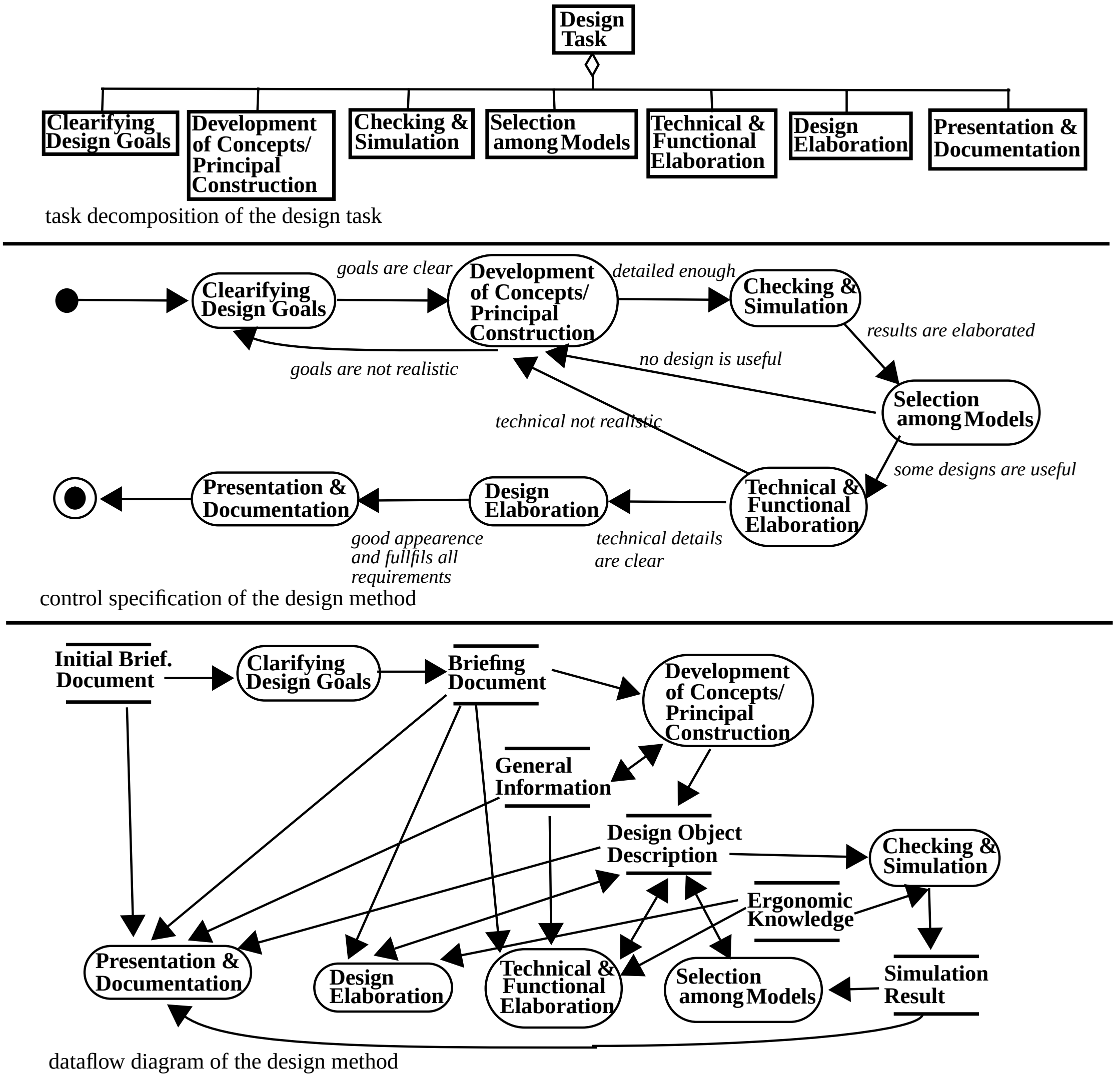

Fig. 10 . Example Process View

for demonstrating the integration of PSMs into an business process and it is simple to extend this CMR version to a full fledged PSM. We first deal with the data view and the domain/case knowledge. We have chosen different concepts, so we have to introduce a mapping association between the concepts of the data view and the concepts of the domain/case knowledge. The mapping association has to perform an ontology mapping between the two different levels.

The task decomposition of the step "Technical & Functional Elaboration" is just the two steps of the PSM CMR: complete design model and revise design model. So in the statechart the two subtasks can be viewed as substates of the "Technical & Functional Elaboration" state.

In the dataflow diagram the dataflow in the expertise view is just a refinement of the dataflow in the process view. The mapping association performs the mapping of the documents of the overall task to concepts usable for the PSM: For every data store in the right dataflow diagram there has to be at least one corresponding (through a mapping association) data store in the left dataflow diagram.

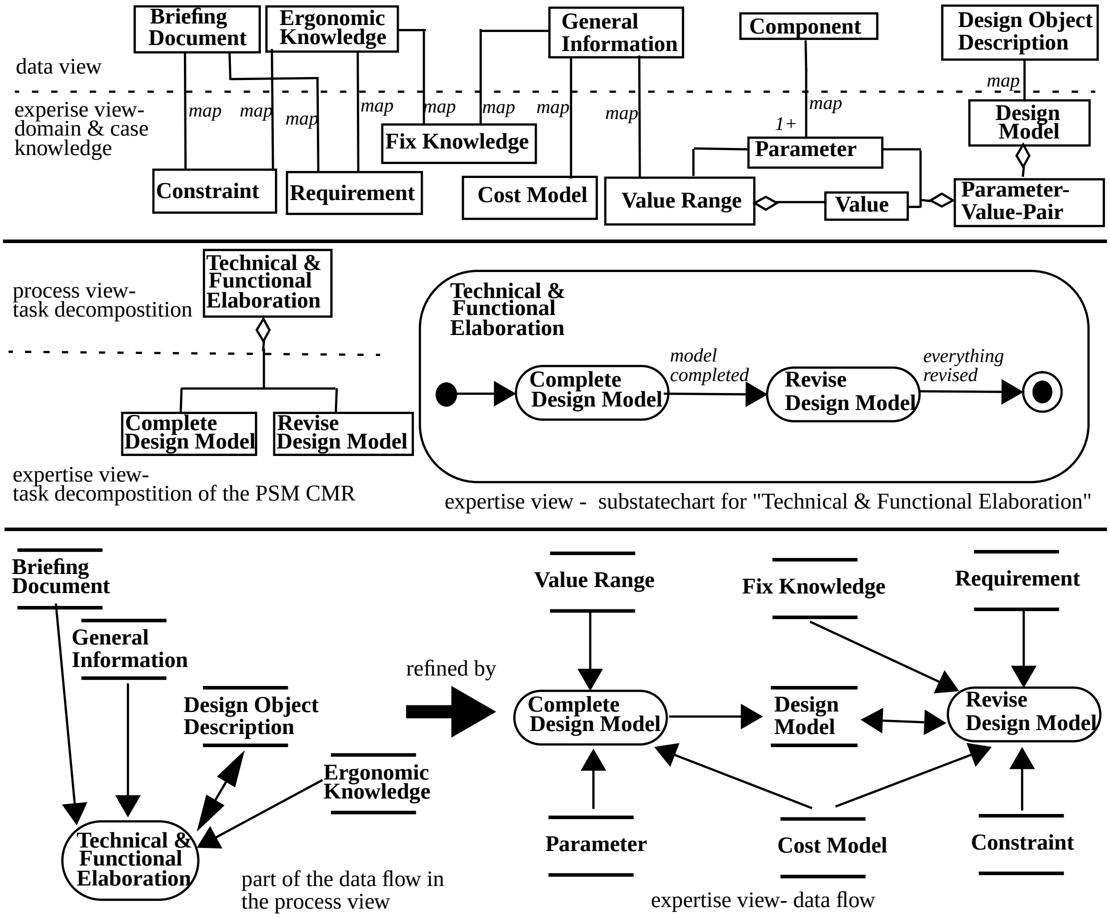

Fig. 11 . Example Expertise View

6 Related Work

The importance of capturing the characteristics of the workplace context in which a KBS should be used is stressed in [VaM94]. This approach proposes a so-called workplace ontology to describe among others the organizational embedding of the system, available resources, and expected problems. However, in contrast to our approach, there does not exist an explicit model of the workflow the KBS is embedded in. I.e. the proposal of Vanwelkenhuysen and Mizoguchi is representing static aspects of a workplace, whereas our approach also takes into account the dynamic aspects of a workplace context.

Another widespread modelling approach (including tool support) suitable for comparison is ARIS ("Architektur integrierter Informationssysteme", integrated information systems architecture [Sch94]). The architecture or basic orientation frame of MIKE and ARIS is given by two dimensions orthogonal to each other. In one dimension, both approaches distinguish distinct *views* on the object worlds to be modelled. The dimension 'degree of formalisation' in MIKE (informal, semiformal, formal) corresponds to the dimension of *levels* in ARIS (application level, data processing concept level, implementation level). Both dimensions refer to increasing formalisation or data processing orientation, respectively. However, ARIS does not consider informal models, so a reference from the semiformal models of the application level to respective primary inquiry information cannot be realised. Relevant modelling aspects for MIKE that are not supported by ARIS are for example the modelling of knowledge (expertise view), qualification profiles of employee groups (staff view), the distribution of tasks (cooperation view), and the communication (communication view) between human and computer.

The reference scheme of [RaV95] is similar to ours: they define a reference scheme with a non standard notation for business modelling and show an operationalisation with high level petri nets. We, however, focus on the standard notation OMT and especially the interface between knowledge engineering (esp. problem solving methods) and business process modelling.

In [KiB94] the notion of an Enterprise Model is introduced. Such an Enterprise Model is composed of several submodels: objectives model, activities and usage model, actors model, concept model, and information systems requirements model. In that way, the Enterprise Model aims at capturing all aspects which are relevant when developing an information system in a business context, i.e. it defines a meta-level framework which specifies the type of knowledge which has to be modelled within each of the submodels. We can interpret our approach as a concrete instance of such a meta-model, i.e. as a proposal of how to represent such submodels and their relationships.

A meta-model approach for modelling business processes is described in [JJP+96]. Jarke et al. propose the definition of a language meta model which can be used to describe different views on business processes. Their proposal for a meta language aims at modelling quality-oriented business processes and puts emphasis a.o. on supporting the negotiation process which is needed to achieve coherent views. On the other hand, their approach does not consider the development of a KBS and does not pay much attention to the persons working in an organisation.

The Brahms-Framework (cf. [CSS+96]) is oriented towards the informal modelling of scenarios in a situated way: every activity of an employee is collected and described in a so called workframe. A workframe contains a semiformal description of an activity which can be further analysed. Also Brahms' models are somewhere between cognitive models and business process models regarded to the level of detail. The main differences to our approach is that we don't focus on informal aspects of an enterprise: instead we model several views of an enterprise aiming at a smooth transition from business process models to problem solving methods using a standard graphical notation from software engineering. Also Brahms is not especially directed to the development of KBS.

The organisation model of CommonKADS [HBM+96] is oriented towards knowledge engineering. The organisation model is constructed when a project has been set up for building a KBS. This is in contrast to our approach which considers the development of a KBS just as one alternative for achieving a business goal. Furthermore CommonKADS does not support graphical modelling in a unified modelling language:

e.g. the process constituent is not very elaborated: no description method is provided to allow a modelling of business processes and to link them explicitly with the model of expertise.

It is not very surprising, that the entities and concepts dealt with in other modelling approaches are very similar to ours. It would be very surprising, if this would not be the case. However, none of the approaches has paid attention to the following: integration of the development of knowledge based systems in a business process reengineering methodology for enterprises. We achieved this by defining a common model for enterprise modelling [DES96], defining a reference scheme usable for many different problems and proposing a common modelling language for the reference scheme, which is usable in enterprise modelling, software engineering and knowledge engineering.

7 Conclusion and Future Work

We defined an enterprise meta model and showed, how it is connected to model-based knowledge engineering: as mentioned above, by using the MIKE approach to model the business views as well, the modelling of the KBS is tightly connected with business modelling. In that way relevant information can be extracted from according views. It is already structured and serves as a reference because of the links established from the model of expertise to the process and data view and to all the other business views. Thus traceability of information or requirements is highly supported by this integration of BPM and KBS development.

The generic process model of [DES96] does not state explicitly how solutions to business problems should look like. These solutions could consist of a KBS, an information system, a workflow engine or any other means of business functions. In this aspect the MIKE approach can be useful in a more generic way: although it is oriented towards building KBS, parts of it can be used as well for developing other kinds of software, i.e. MIKE could be viewed as the basis for a general requirements modelling and system specification method for knowledge management systems.

8 References

[Ang93] J. Angele: Operationalisierung des Modells der Expertise mit KARL (Operationalization of the Model of Expertise with KARL), Ph.D. Theses in Artificial Intelligence, No. 53 , infix, St. Augustin, 1993.

[AFS96] J. Angele, D. Fensel, and R. Studer: Domain and Task Modeling in MIKE. In: Proceedings of the IFIP WG 8.1/ 13.2 Joint Working Conference, Domain Knowledge for Interactive System Design, Geneva, May 1996.

[BKM94] B.M. Bauer, C. Kohl, H.C. Mayr, J. Wassermann: Enterprise Modeling using OOA Techniques,In: Proceedings Connectivity '94: Workflow Management - Challenges, Paradigms and Products. München: R. Oldenbourg-Verlag, 1994, pp. 96-111.

[Bro86] A.G. Brooking: The Analysis Phase in Development of Knowledge-Based Systems. In: A. Gale (ed.): AI and Statistics. Addison-Wesley Publ. Co., 1986.

[CSS+96] W. J. Clancey, P. Sachs, M. Sierhuis, R. van Hoff: Brahms: Simulating Practice for Work Systems Design, Proceedings of PKAW'96.

[DaF92] M. Daniel, G. Fieguth: Möglichkeiten der kontrastiven Aufgabenanalyse auf die Entwicklung wissensbasierter Systeme. Forschungsbericht, ibek GmbH, Karlsruhe 1992

[Dan93] M. Daniel: Formen der Unterstützung von Diagnosetätigkeiten, ibek GmbH, Karlsruhe 1993

[DES96] S. Decker, M. Erdmann, R. Studer: A Unifying View on Business Process Modelling and Knowledge Engineering. In: Proceedings of the 10th Knowledge Aquisition Workshop (KAW 96), Banff, Canada, November 1996.

[DuV92] H. Dunckel et. al: Kontrastive Aufgabenanalyse. Das KABA-Verfahren. Verlag der Fachvereine, Zürich 1992.

[Fen95] D. Fensel: The Knowledge Acquisition and Representation Language KARL, Ph. D. Thesis, University of Karlsruhe, 1993. Kluwer Academic Publisher, Boston, 1995.

[FAS97] D. Fensel, J. Angele, R. Studer: The Knowledge Acqusition und Representation Language KARL. to appear in IEEE Trans. on Knowledge and Data Engineering, 1997

[HBM+96] R. de Hoog, B. Benus, M. Vogler, C. Metselaar: The CommonKADS Organisation Model: Content, Usage and Computer Support. In: Journal of Expert Systems with Applications, Vol.11, No. 1, pp.29-40, 1996

[JJP+96] M. Jarke, M.A. Jeusfeld, P. Peters, and K. Pohl: Coordinating Distributed Organizational Knowledge. To appear in: Data & Knowledge Engineering, 1997

[KJO+95] H. Kangassalo, H. Jaakkola, S. Ohsuga, B. Wangler (eds): Information Modelling and Knowledge Bases VI, IOS Press, 1995

[KKM95] R. Kaschek, C. Kohl, H. C. Mayr: Cooperations - An Abstraction Concept Suitable for Business Process Re-Engineering. In: J. Györkös, M. Krisper, H.C. Mayr (eds.): ReTIS'95 - Re-Technologies for Information Systems. OCG Lecture Notes 80, Oldenbourg, 1995.

[KiB94] M. Kirikova and J.A. Bubenko: Software Requirements Acquisition through Enterprise Modeling. In: Proc. 6th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE´94), Jurmala, 1994.

[Lan94] D. Landes: DesignKARL - A language for the design of knowledge-based systems. In: Proc. 6th International Conference on Software Engineering and Knowledge Engineering SEKE'94, Jurmala, Lettland, 1994, 78-85.

[LaS95] D. Landes and R. Studer: The Treatment of Non-Functional Requirements in MIKE. In Wilhelm Schäfer, Pere Botella (eds.): Proceedings of the 5th European Software Engineering Conference ESEC'95 (Sitges, Spain, September 25 - 28, 1995), Berlin, Springer 1995 (Lecture Notes in Computer Science; 989).

[MoZ96] E. Motta, Z. Zdrahal: Parametric Design Problem Solving. In: Proceedings of the 10th Knowledge Aquisition Workshop (KAW 96), Banff, Canada, November 1996.

[Neu93] S. Neubert: Model construction in MIKE (Model-based and Incremental Knowledge Engineering). In: N. Aussenac et.al. (eds): Proceedings of the 7th European Workshop EKAW'93 (Toulouse), LNAI 723, Springer-Verlag, Berlin, 1993.

[Neu94] S. Neubert: Modellkonstruktion in MIKE - Methoden und Werkzeuge. Ph.D. Thesis in Artificial Intelligence, No. 60, infix, St. Augustin, 1994 (in German).

[Neb96] B. Nebel, Artificial Intelligence: A Computational Perspective. In: G. Brewka (ed.): Essentials in Knowledge Representation. Studies in Logic, Language and Information, CSLI Publications, pp. 237-266, 1996

[RaV95] G.J. Ramackers, A.A. Verrijn-Stuart: Conceptual Model Requirements for Integrated Business and Information System Development. In:[KJO+95], pp.: 197-213

[RBP+91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen: Object-Oriented Modeling and Design. Prentice Hall, Englewood Cliffs 1991.

[Sch94] A.-W. Scheer: Business Process Engineering: Reference Models for Industrial Enterprises, Springer, 2nd ed. 1994.

[ScB96] A.Th. Schreiber and B. Birmingham (eds.): International Journal of Human-Computer Studies (IJHCS), Special Issue: The Sisyphus-VT Initiative, 1996.

[SWB93] G. Schreiber, B. Wielinga, and J. Breuker (eds.): KADS. A Principled Approach to Knowledge-Based System Development, Knowledge-Based Systems, vol 11, Academic Press, London, 1993.

[ScW93] G. Schreiber, B. Wielinga: Model Construction. In [SWB93], pp 93-118..

[ScW93a] G. Schreiber, B.J. Wielinga: KADS and Conventional Software Engineering. In: KADS - A Principled Approach To Knowledge Based System Development, Academic Press, 1993

[SWH+94] A.Th. Schreiber, B.J. Wielinga, R. de Hoog, H. Akkermans, and W. van de Velde: CommonKADS: A Comprehensive Methodology for KBS Development. In: IEEE Expert, December 1994, 28-37.

[ShG92] M.G. Shaw and B.R. Gaines: The Synthesis of Knowledge Engineering and Software Engineering. In: P. Loucopoulos (ed.): Advanced Information Systems Engineering, LNCS 593, Springer Verlag, 1992.

[SEG+96] R. Studer, H. Eriksson, J. Gennari, S. Tu, D. Fensel, M. Musen: Ontologies and the Configuration of Problem-Solving Methods. In: Proceedings of the 10th Knowledge Aquisition Workshop (KAW 96), Banff, Canada, November 1996.

[VaM94] J. Vanwelkenhuysen and R. Mizoguchi: Maintaining the Workplace Context in a Knowledge Level Analysis. In: Proc. 3rd Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop(JKAW´94), Hatoyama, 1994