A Unifying View on Business Process Modelling and Knowledge Engineering[1]

Stefan Decker, Michael Erdmann, and Rudi Studer
Institut für Angewandte Informatik und Formale Beschreibungsverfahren
University of Karlsruhe (TH)
D-76128 Karlsruhe (Germany)
e-mail: {decker | erdmann | studer}@aifb.uni-karlsruhe.de

Abstract

Recently the demand on business process modelling (BPM) became apparent in many different communities. To provide a unifying framework we propose an approach to integrate (well known) knowledge engineering techniques into a business process modelling context. We see knowledge based systems as one possibility (among others) to implement (re-engineered) business processes or parts thereof. Our framework is exemplified by integrating the MIKE approach (Model based and Incremental Knowledge Engineering) with ARIS (ARchitecture of Information Systems), a prominent representative of business process modelling. An important aspect of our framework is a proposal for linking business process models with a model of expertise.

1 Introduction

Recently the demand on business process modelling (BPM) became apparent in many different communities, such as workflow management [Österle 95], information systems engineering, requirements engineering [Kirikova and Bubenko 94], software engineering and knowledge engineering (e.g. [Breuker and Van de Velde 94], [Schreiber et al. 94]). This suggests a unifying view on business process modelling and the other cited areas. 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 process the whole business process including organizational structures and activities has to be modelled. This is also true when building a KBS in an enterprise environment. Because the KBS represents only a small part of the whole business organization it must be embedded or at least linked to all relevant business processes, i.e. it should not be a stand-alone solution.

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

We propose to combine solutions derived by knowledge engineering (KE) and BPM. The observation of this close relationship is not new, e.g. the *CommonKADS* organization model [de Hoog et al. 94] defines several components, that also arise in common business modelling approaches. In Fig. 1 we present our view of the development process of business process models and their improvement. This spiral model is not intended to suggest a specific process, but should instead present the relations between business process models and any kind of improvement. It makes clear how to embed the development of KBSs into the BPM process. To solve business problems one alternative is to develop software of any kind, among which a KBS is only a sub-alternative. On the other side the development of a business process model can be seen as a knowledge engineering activity by itself, so that methods from the area of knowledge engineering are also applicable to business process modelling.

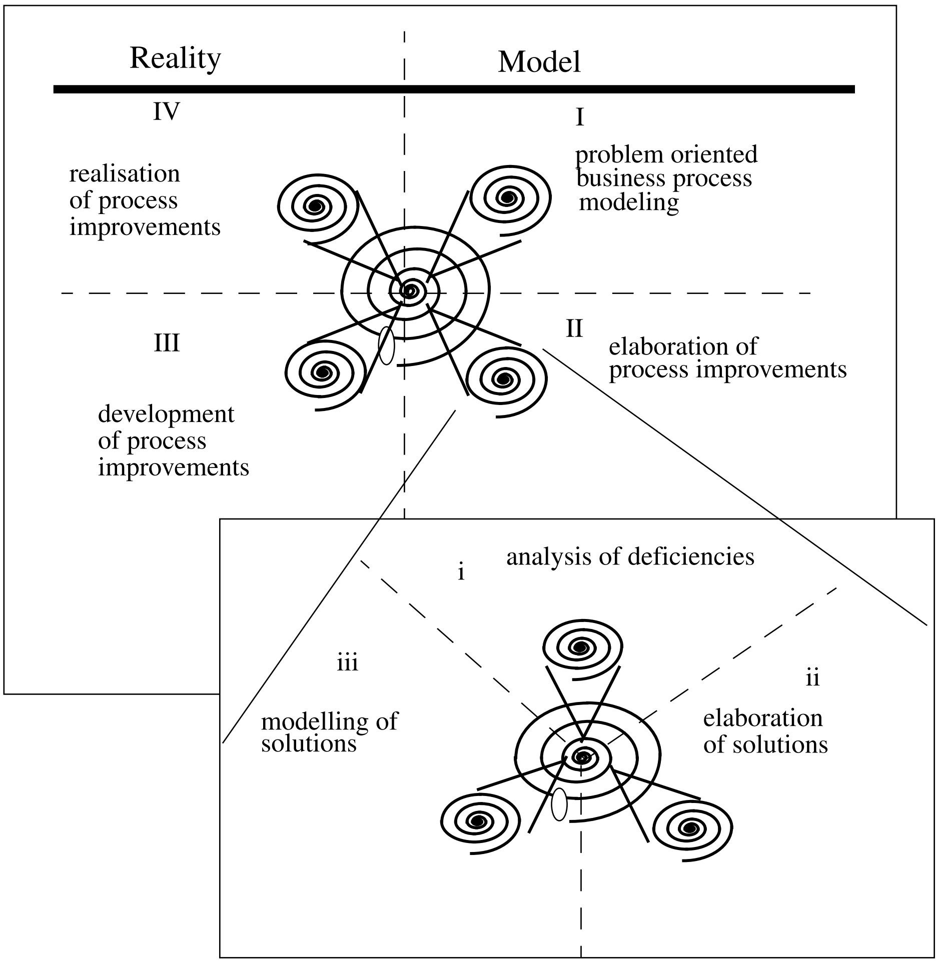

Fig. 1 Generic Process Model

Because the spiral model and its instances (e.g. ARIS [Scheer 92]) do not concentrate on developing knowledge based systems the following question arises:

1. "Which aspects must be added to a business process modelling methodology if it should support *the enactment of a KBS?"*

We are following the KADS approach [Schreiber et al. 93] which states that the main constituent of a KBS is the model of expertise (i.e. a model of the problem solving knowledge). The central point of this paper is to address the integration of knowledge based systems and business process models, thus answering the question:

2. *"How are the business process models and the model of expertise represented and in what way should they be linked or how do they interact?"*

The remainder of the paper is organized as follows: In section 2 we will describe the business process approach based on ARIS (ARchitecture of Information Systems, [Scheer 92], [Scheer 94]). In section 3 we will briefly describe our MIKE approach (Model based and Incremental Knowledge Engineering, [Angele et al. 96a]) to knowledge engineering. Section 4 describes our unified view of KBSs and business process models in general. This description is based on a detailed analysis of ARIS and MIKE. We will present some related work in section 5 before we will close with some conclusions.

2 Business Process Modelling

Several business process modelling approaches are known: e.g. ARIS (Architecture of Integrated Information Systems, cf. [Scheer 94] and [Scheer 92]), INCOME/STAR [Oberweis et al. 94], etc. We have selected ARIS as a basis for developing our unifying view, because:

- ARIS has an elaborated decomposition of an enterprise in several views: data view, function view, organization view and, to realize the connection between these views, the control view (cf. Fig. 2).
- ARIS integrates several modelling methods, e.g. entity relationship models and object oriented approaches, e.g. OMT [Rumbaugh et al. 91].
- ARIS is supported by a toolset which is commercially successful and used in several industrial applications.

The concept of ARIS is based on different views on an enterprise. In detail, ARIS distinguishes between four different views which are described in three layers of abstraction (cf. Fig. 2).

1. In the **organization view** the relations between enterprise units and their classification into the organizational hierarchy are modelled.
2. The **data view** describes objects, their attributes and inter-object relations. Furthermore the data view contains events, that can initiate and control processes.
3. The **function view** embodies functions, that are part of processes and determined through the creation and change of objects and events. A complex function can be decomposed in more elementary ones.
4. The task of the **control view** is besides the integration of the first three views the definition of the dynamic aspects. The most important entities here are functions and events which are linked together to form the so called *event driven process chain* (EPC). The EPC models the control flow of

the business process. It can be (and usually is) extended by links to other relevant entities contributed by other views. So functions can be connected to their input and output data which are located in the data view to model the data flow.

As mentioned above each of these views is described in different levels of abstraction. The starting point is always the managerial/economic description of the enterprise domain, the **requirements definition**. These concepts are formulated in business terms but are strongly influenced by technical possibilities. The resulting models of this first level are laid down in semiformal diagrams. The **design specification** is also modelled semiformally, but uses terms of the envisioned information systems solution (i.e. it speaks about modules and transactions). The last part consists of the physical **implementation description** of the upper levels.

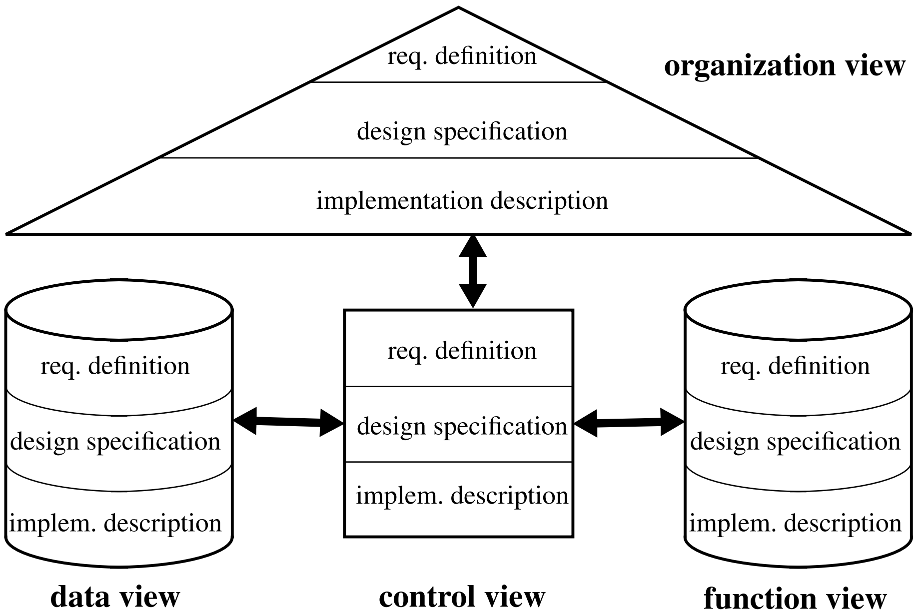

Fig. 2 The ”ARIS house“ [Scheer 92]

3 The MIKE Approach

MIKE (Model based and Incremental Knowledge Engineering, [Angele et al. 96a]) 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 in general *cyclic*, *faulty* and *approximative*.

Within the MIKE approach these properties have been considered as the main reasons in order to develop methods and techniques which provide feedback for the knowledge engineer as early as possible within the modelling process. Therefore prototyping of the acquired expertise using executable models has been integrated into the modelling process as one of its main features.

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* (cf. Fig. 3 and [Neubert 93]). This knowledge which is represented in natural language must be interpreted and structured. The result of this step is described semiformally in the so-called *structure model* [Neubert 93], using predefined types of nodes and links. The structure model consists of four contexts for capturing functional aspects: 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* [Schreiber et al. 93]. For describing the model of expertise the formal and operational specification language KARL (cf. [Fensel 95], [Fensel et al. 97]) has been developed. (An extended version of KARL is described in [Angele et al. 96b].) 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 the KADS model of expertise, respectively.

The model of expertise includes all functional requirements of the desired system. For the realization of the final system, non-functional requirements, such as efficiency of the problem-solving method, maintainability of the system or persistency of data etc. have to be considered [Landes and Studer 95]. 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 [Landes 94] which allows to describe data structures, algorithms and offers additional structuring primitives like clusters and modules.

The different representations of knowledge, i.e. elicitation model, structure model, model of expertise, design model and the final system either represent the same knowledge in a different way or contain additional knowledge which is closely related to other knowledge in another representation. To gain the full benefits for documentation, maintenance, and explainability these different models are interrelated explicitly. Thus traceability of the system development process is achieved.

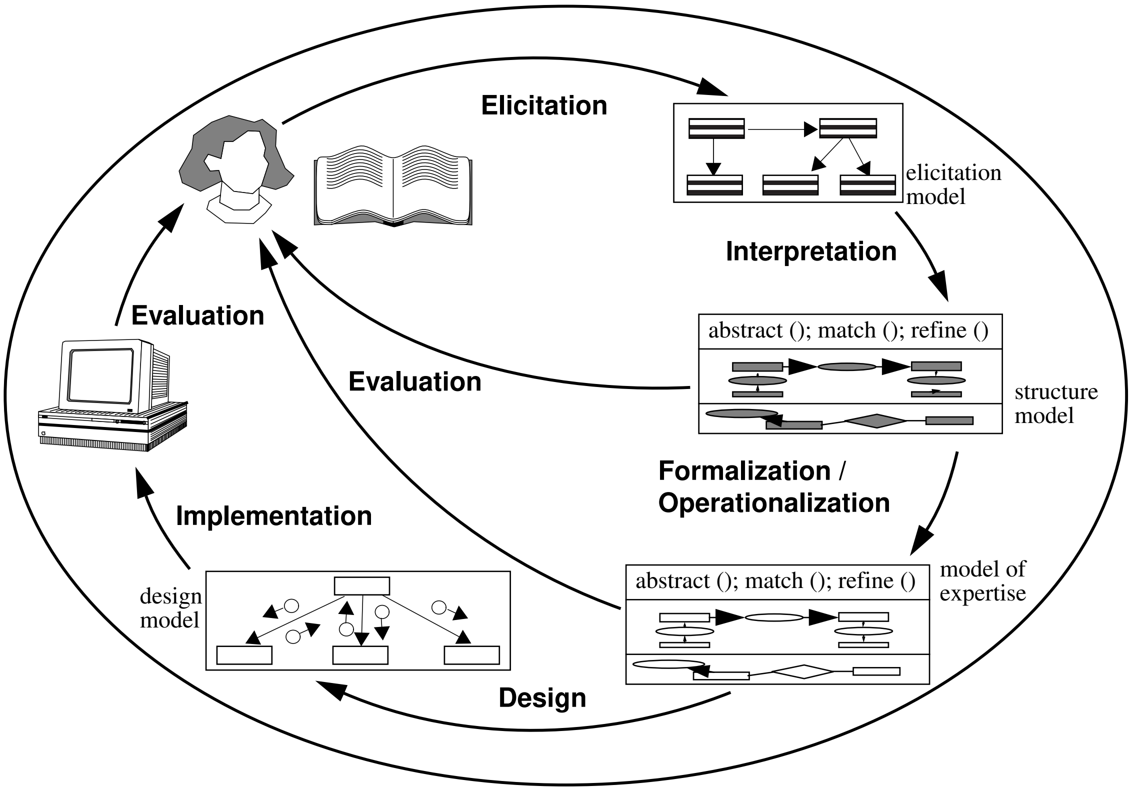

Fig. 3 Representation Levels in MIKE

The different models are results of different steps of the building process. For this process a spiral process model [Boehm 88] has been defined (cf. [Angele 93], [Neubert 93]), which describes the different activities, their resulting models and the sequence of the activities within the whole building process. Due to the above mentioned properties of this modelling process explorative and experimental prototyping have been integrated [Floyd 84] as a main feature.

For the design process the language DesignKARL additionally allows to describe the design process itself and to describe interactions between design decisions. Thus, the design process is documented and the maintainability of the final KBS is improved.

4 Aspects of Combining BPM and KE

After presenting one prominent BPM approach and a KE approach, we will develop a unified view on both in the following section. For this combination at least four different postulations have to be fulfilled:

1. The conceptual aspect: relevant concepts (e.g. data, activities), that are comparable in both approaches, have to be identified.

2. The different levels of abstraction/specification in both approaches have to be considered and a common view has to be developed.
3. The notational aspect has to be defined in such a way, that it is possible to create a smooth transition from one approach to the other.
4. It is necessary to integrate the life cycle models of both approaches to achieve a practicable, unified, new process model (cf. Fig. 1).

To identify the relevant comparable concepts, we use the perspectives which are "generally recognized as relevant for both business modelling and information system modelling" (cf. [Ramackers and Verrijn-Stuart 95], p.199) and thus also for modelling KBSs. [Ramackers and Verrijn-Stuart 95] "distinguishes the *data* (or 'structural', 'information') perspective, the *process* (or '*functional*', 'data flow') perspective and the *behaviour* (or 'event', 'control') perspective" (cf. Fig. 4).

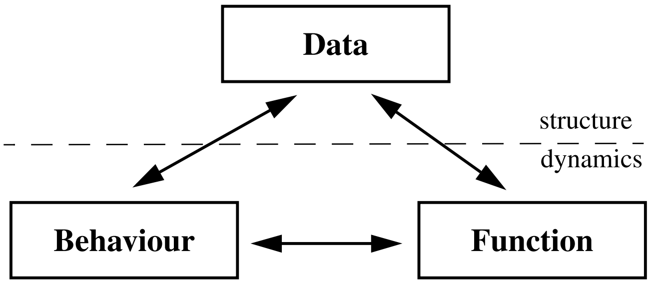

Fig. 4 Model Perspectives (cf. [Ramackers and Verrijn-Stuart 95], p.199)

Object oriented modelling has proved to be able to describe all relevant perspectives, e.g. the Object Modelling Technique (OMT, cf. [Rumbaugh et al. 91]) contains exactly the above mentioned perspectives. In OMT they are called the *object model*, the *functional model*, and the *dynamic model,* resp. OMT is well suited for BPM and KE, so we will present a unified view on ARIS and MIKE by defining relationships between corresponding models and exploiting the common notation of OMT.

4.1 Levels of Specification

ARIS as well as MIKE propose different levels of abstraction or specification. In MIKE these levels are characterised by their notation as well as by their contents [Angele et al. 96a]. During the knowledge elicitation phase informal protocols are produced, constituting the so called *elicitation model*, which contains e.g. natural language texts or other hypermedia sources. These protocols are formulated in domain dependent terms to maintain a communication basis between domain experts, users and knowledge engineers. The second layer of specification in MIKE (the *structure model*) is constructed by interpreting the informal protocols and identifying relevant concepts and activities. The knowledge engineer produces thereby semiformal diagrams (OMT object models to represent the static domain characteristics; state transition diagrams and data flow charts to represent the dynamic aspects). These semiformal models are linked via hyperlinks with corresponding parts of the protocols for documentation and traceability purposes. In the subsequent formalization step the formal *model of expertise* (specified in KARL) is constructed from the semiformal structure model. This model of expertise serves as an executable prototype. Design decisions taking non-functional requirements into account result in the *design model* (described in DesignKARL) which can be implemented to yield the knowledge based system.

As mentioned in section 2 ARIS focuses from the start on developing an information system. This leads to relatively technical descriptions even on the *requirements definition* level. In this layer relevant data, functions and organizational structures are formulated in semiformal diagrams and linked together by the central data structure, the event driven process chain (EPC). Formalisms used in the requirements definition are entity relationship diagrams or OMT object models for the data view, EPCs for the control view and hierarchical structures for functions and enterprise units. The second ARIS level, the *design specification*, speaks about modules, database tables, and transactions. They are also described semiformally, e.g. the control flow of the function view can be modelled using Nassi/Shneiderman-Diagrams. The *implementation*, i.e. the last level of an ARIS description, contains coded programs, implemented database applications etc., i.e. software components.

The ARIS levels of specification are not intended to increase the degree of formalization as are the layers in MIKE. In MIKE the gap between informal specification and final implementation is bridged by several intermediate levels, so that transitions between levels can be achieved relatively easily. In ARIS the gaps between the different levels are still rather large. A striking observation is that ARIS proposes to start with semiformal descriptions whereas the elicitation phase of MIKE requires informal input which can be provided and understood by experts and users more easily.

To present the common view on the different levels of specification in ARIS and MIKE it can be stated that:

- ARIS' requirements definition corresponds roughly to MIKE's structure model;
- the level of specification expressed in ARIS' design specification and in MIKE's design model are equivalent;
- the implementation level of ARIS naturally corresponds to the implemented knowledge based system.

As one can see, in ARIS there are no corresponding levels for MIKE's informal elicitation model or for MIKE's model of expertise. The requirements definition is biased by information system concepts. By this it is obvious that ARIS is directly heading towards a computational implementation.

Besides this *vertical* comparison, the two approaches can be compared *horizontally* as well, i.e. the different views of ARIS can be compared with MIKE's components of the structure model (concept, activity, data flow, and ordering context) or the model of expertise (the domain, inference, and task layer). This comparison and a proposal to unify the components will be given in the following subsections. Because in MIKE the transition between structure model and model of expertise is structure preserving, we could compare ARIS' three perspectives with either MIKE's structure model or its model of expertise. The reason why a comparison with the structure model is more appropriate is, that both ARIS' requirements definition and MIKE's structure model are semiformally described. A comparison on a lower level of abstraction seems less adequate.

4.2 The Static Views

In ARIS the static aspects are scattered over two views, namely the data view and the organization view. The latter contains hierarchical diagrams which represent the enterprise units and their relations (usually instruction relationships). The data view contains classes, objects, their attributes and relationships relevant for the business processes. In MIKE these entities are modelled in the concept context of the structure model. Both approaches utilize ER diagrams or OMT to model the static aspects. The model of the organizational environment in ARIS provides information that is currently not represented in any of the MIKE models. However, [Decker et al. 96] describe how MIKE will be extended in this way.

The problem solving process modelled in MIKE reasons about entities appearing in an expert's problem solving process. Therefore they are modelled in MIKE's concept context. The entities modelled in ARIS' data view are identified by their business process context, and thus are primarily physical objects. But there is of course a close relationship between these two concepts, because the experts reason about real world objects. So many objects appear in MIKE's concept context as well as in ARIS' data view. We propose to unify these two views to enable a smooth transition from enterprise modelling to KBS modelling. This can be done in two ways. Firstly, the link between the two views can be modelled by convention, i.e. the same objects must be named equally. Or secondly the link must be explicitly defined, e.g. by defining an interface. The argument supporting the second solution is as follows: concept context and data view may be developed independently (one part may even be reused) so that an explicit mapping between objects of both parts is necessary. The first choice forces the developers to reflect the organizational environment, esp. the used objects that should further be used to specify the static aspects of the KBS. Therefore we propose to use the first solution to achieve a homogeneous view of objects.

As representational formalism for the data and organization view and the concept context we use the OMT object model (cf. Fig. 5), because of its wide dissemination. I.e. classes can be defined with associated attributes. Between these classes relationships can be established (generalisation, aggregation, or any other association) which can be annotated by attributes, too (cf. Fig. 5).

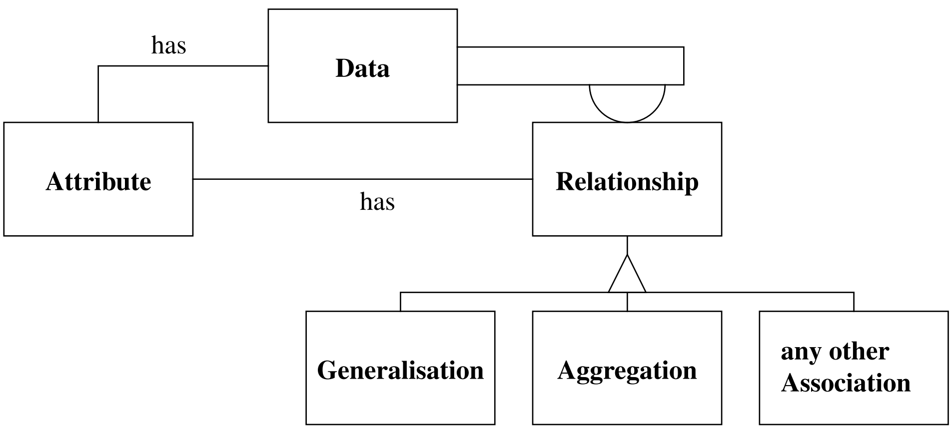

Fig. 5 Meta Model for the Static Views

4.3 The Dynamic Views

As mentioned at the beginning of section 4, dynamic aspects are described by behavioural and functional perspectives. In MIKE the behaviour perspective is modelled by a hierarchical task decomposition tree (the activity context) and control flow diagrams which define the ordering of activities (in the ordering context). ARIS models its function view in the same way. It contains a decomposition tree of business processes and an ordering of subprocesses (modelled by restricted EPCs or Nassi/Shneiderman-Diagrams).

There is no unequivocal way to perform a given task. Several processes may lead to a solution for a task. These processes may decompose the original task into more elementary subtasks, thus leading to a set of tasks which are solved by a set of more elementary processes etc. until atomic processes are reached which cannot be further decomposed. The distinction between processes and tasks is relevant in business processes as well as in problem solving processes [Schreiber et al. 94]. We have developed a new meta model (cf. Fig. 6) which addresses this important observation. The process view (cf. Fig. 6a) states that each process (in an enterprise) consists of the definition of data and control flow. Thus, the process model combines the behaviour *and* the functional perspective. It says, that there are several levels of business processes/tasks (typically three). The first level contains tasks which involve several enterprise units or employees, the second level contains tasks which are handled by a single person, and the third level describes single actions performed by that employee.

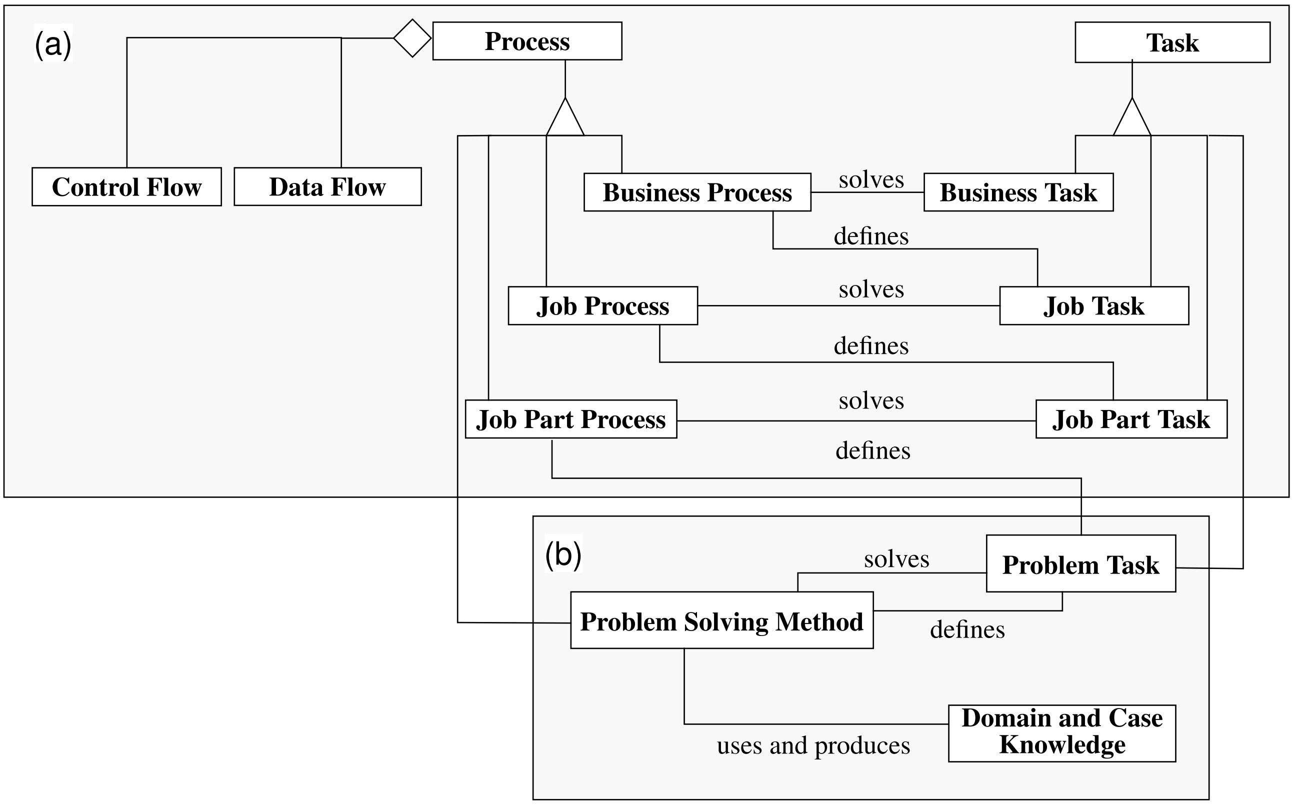

Fig. 6 Meta Model for (a) Process View and (b) Expertise View

Because problem-solving tasks and processes are related in the same way as above, the meta model describing these dynamic aspects (cf. Fig. 6b) looks much the same. In this expertise view it is not reasonable to identify a concrete number of decomposition steps. So the model states that a problem solving method solves a certain problem task by defining subtasks which are solved by submethods etc. without restricting the number of description levels [Eriksson et al. 95].

The difference between business processes on the one hand and problem solving activities on the other hand has hardly any effect on process structuring, i.e. both, business and problem solving processes can be ordered in a hierarchical as well as in a sequential manner. The problem solving process (of an expert) is embedded in the business process. So it can be viewed as a (fine grained) business process on its own. This perception is modelled via inter-view-links, connecting the expertise view with the process view. Two links define is-a relationships between a problem-solving method and a process as well as between a problem task and a task. Another link relates job part process and problem task. This relation is central and states that a job part process can be decomposed into a set of problem solving tasks, thus establishing the transition between business processes and problem solving processes.

After defining the task/process decomposition of business and problem solving processes and linking the two task/process types together we will relate the processes with the static aspects of a model, i.e. the data and the organization views. As we have seen, a (business or problem solving) process contains a data flow part and is thus connected to the data view. Currently, in MIKE all static aspects are represented in the concept context, thus defining a data flow diagram is sufficient to relate activities and concepts. On the other hand ARIS and our unified view contain a second static aspect, the organizational view, which must be linked to the processes to provide useful information. Confer [Decker et al. 96] for current extensions of MIKE with respect to organizational aspects.

The central component of ARIS is the control view, which links together all other ARIS views (cf. Fig. 2). Within the control view functions/processes and organizational units (employees) are associated. The mentioned units are capable of or responsible for executing processes. In a KBS context these units or employees can serve as knowledge sources because of their abilities concerning the business process, i.e. their knowledge may lead to a problem solving process, which can be implemented by a KBS.

Both, MIKE and in ARIS use their own methods for describing the behaviour and functional perspectives. To be able to integrate business processes and problem solving activities it is necessary to utilize a uniform notation. Because a main difference between MIKE and ARIS is the use of concurrency in EPCs (a business process is often naturally divided into several concurrent subprocesses), a synchronisation between these subprocesses is necessary [Kaschek et al. 95]. Although the specification of parallel problem solving methods is not yet supported in the knowledge engineering community, we extend the MIKE formalisms by concurrency and synchronisation facilities, using OMT's dynamic model (i.e. state charts). Thus one notation serves in an appropriate way for the behaviour perspective of both approaches (cf. [Bauer et al. 94], [Schreiber and Wielinga 93] p.162).

For modelling the functional perspective we propose to use a kind of data flow diagrams, both for business processes (as EPCs do) and problem solving activities (as in MIKE). By that, we achieve a single representation and thus can model the whole dynamic perspectives of business processes and knowledge based systems alike. It allows to specify a smooth transition from business to problem solving processes.

4.4 Survey of Relevant Views

After describing an unifying view on the enterprise models and MIKE's structure model we will give an overview about all relevant views we identified[2] and their (coarse grained) relationships, even if we do not further explain most of them in this paper.

Models (or views) generally have the objective to simplify complex realities by representing only relevant aspects. The selection of the views in Fig. 7 is orientated towards similar views in ARIS and MIKE: The set of relevant models therefore can be separated into two subsets. The first subset consists of the conventional BPM views. For enterprise modelling, the distinction between the structure (*organizational view*) and the processes of an organization (*process view*) is essential. In addition, for building new processes, the resources available in an enterprise are relevant. We model these resources in the *staff view* and the *working tool view.* This distinction has to be made due to human demands on their workplaces. The *data view* is a standard view on organizations, because data is exchanged between different tasks. Communication and cooperation are important for the reengineering of processes (e.g. task splitting between human and computer) and are therefore treated as a separate view (*communication and cooperation view*).

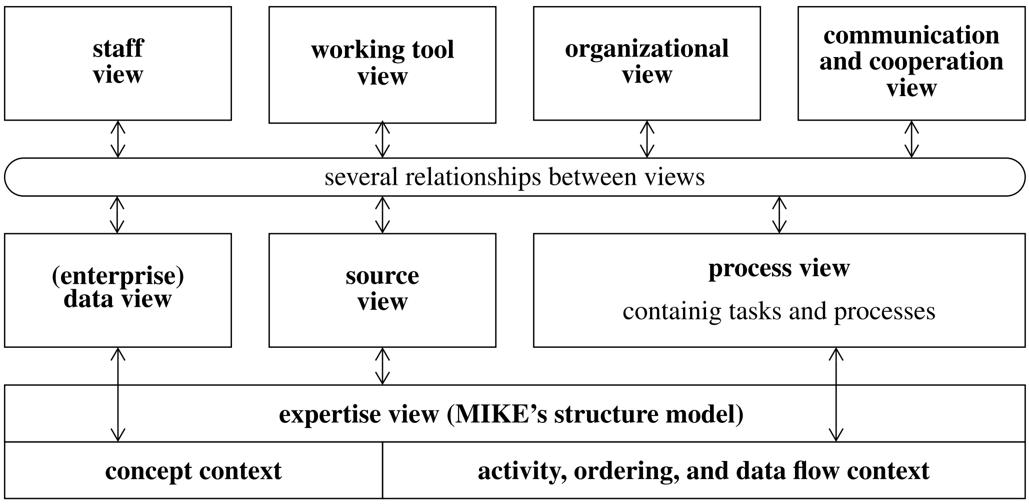

Fig. 7 Survey of Relevant Views

2. The given models result from research in the ongoing WORKS project (Work oriented Design of Knowledge Systems). For more details confer [Decker et al. 96].

The second subset consists of views that are used in model based knowledge engineering: the *expertise view* is founded on the MIKE approach on knowledge-based systems (cf. section 3). In connection with this, the usable knowledge sources (*source view*) are interesting for purposes of knowledge acquisition. The connection between these views has been discussed in section 4.3.

The business modelling process is typically started because certain problems arise which obstruct 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 to state how to solve these problems. If the decision leads to constructing a KBS then the second subset becomes relevant. The expertise view provides all necessary information when building a KBS. As one can see (cf. Fig. 7) certain views of both subsets are connected via links. These links tie together the business modelling and the development of a suitable KBS. By filling the expertise view, probably further information must be elicited to model the problem solving behaviour, so further protocols are produced. These protocols as well as the already defined business views serve as input for the expertise view. In that way the higher level business views are closely connected with the expertise view.

5 Related Work

5.1 CommonKADS

CommonKADS [Schreiber et al. 94] is a comprehensive methodology for KBS development. The methodology proposes especially the development of an organization model [de Hoog et al. 94]. Although this organization model is elaborated and contains many constituents, the *CommonKADS* model set focuses (only) on the development of KBSs. I.e. no integrated approach is described to use these models in a general information systems or business process environment. The organization model considers dynamic aspects, such as workflow, only in one constituent (its process constituent), i.e. its focus lies on representing the static aspects holding in an enterprise. The defined set of models does not provide a direct integration of the *CommonKADS* model of expertise with parts of the organization model. Instead there exists only an implicit link between the process constituent in the organization model and the expertise model via the task model. The necessity of explicitly linking business process models and the model of expertise is not stated in *CommonKADS*.

5.2 Spark, Burn, Firefighter

The SBF (Spark, Burn, Firefighter) project [Yost et al. 94] recognizes the importance of what they call the *task context*. They initially introduce ”models of workplaces and work processes involved in a task [...] to facilitate communication with the task developer“ (the domain expert). In SBF v2 a hierarchy of fix industry models and process models is presented to describe generic business processes. To these business process models so-called mechanisms are tied such that a process model is configured simply by selecting certain business processes. The third version of SBF ”allows the developer to enter a process model from scratch“. By doing this the developers do not solely specify a KBS, instead ”they were describing the cooperative workplace interactions to perform a task“. In effect ”SBF v3 focuses on how to build a cooperative multi-agent human-computer system that effectively performs a task“, i.e. SBF evolved from being a KBS to a workflow management system. However, in contrast to the approach we present in this paper the SBF framework does not provide a collection of well related disjoint models on different abstraction levels.

For associating the elements of a process model with the required mechanisms, which are stored in a library, SBF uses the Active Glossary [Klinker et al. 93]. In essence, this glossary relies on a mapping of the terminology as used in the process model to the terminology as used for specifying the mechanisms. Although this Active Glossary provides some help in handling the indexing problem we do not believe that a simple keyword matching is sufficient for identifying the appropriate mechanisms. Rather, a functional characterisation of the stored mechanisms has to be taken into account as well (cf. [Angele et al. 96b]).

5.3 Other Enterprise Modelling Approaches

The importance of capturing the characteristics of the workplace context in which a KBS should be used is stressed in [Vanwelkenhuysen and Mizoguchi 94]. 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.

In [Kirikova and Bubenko 94] 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.

6 Conclusions

In this paper we have presented a framework for integrating methods of a model-based knowledge engineering approach (namely MIKE) with a business process modelling approach (ARIS). We identified the similarities and differences between these two approaches by using the three model perspectives (data, behaviour and function) as a guideline. The dynamic and the static aspects of both methodologies are comparable, but not identical. To achieve an integration we defined meta views of these aspects and connected them with appropriate links. To this extent we integrated the development of a knowledge based system in a general business process modelling approach. To provide a useful integration of both methodologies we proposed the use of OMT for modelling the different views (cf. [Bauer et al. 94], [Schreiber and Wielinga 93]). Within our framework it is possible to view on the one hand a business process as a kind of problem solving process; on the other hand a KBS may be appropriate to implement such a business process.

We are currently extending our MIKE approach by these views in order to be able to support the integration of the development of KBSs into an enterprise modelling environment [Decker et al. 96].

Acknowledgement.
Fruitful discussions with Manfred Daniel are gratefully acknowledged.

7 References

[Angele et al. 96a] J. Angele, D. Fensel, and R. Studer: Domain and Task Modelling in MIKE. In: A.G. Sutcliffe, D. Benyon, and F. van Assche (eds.): *Domain Knowledge for Interactive System Design. Proceedings of the TC8/WG8.2 Conference on Domain Knowledge in Interactive System Design*, Geneva, Switzerland, May 1996. Chapman & Hall, London, 1996.

[Angele et al. 96b] J. Angele, S. Decker, R. Perkuhn, and R. Studer: Modelling Problem-Solving Methods in New KARL. In: *Proceedings of the 10th Knowledge Acquisition Workshop (KAW'96)*, Banff, Canada, 1996.

[Angele 93] J. Angele: Operationalisierung des Modells der Expertise mit KARL (Operationalization of the Model of Expertise with KARL), Ph.D. Thesis in Artificial Intelligence, No. 53, infix, St. Augustin, 1993 (in German).

[Bauer et al. 94] M. Bauer, C. Kohl, H.C. Mayr, and 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.

[Boehm 88] B.W. Boehm: A Spiral Model of Software Development and Enhancement. *Computer 21*, 5 (May 1988), 61-72.

[Breuker and Van de Velde 94] J.A. Breuker and W. Van de Velde (eds.). *The CommonKADS Library for Expertise Modeling*. IOS Press, Amsterdam, 1994.

[Decker et al. 96] S. Decker, M. Daniel, M. Erdmann, and R. Studer: An enterprise reference scheme for the integration of model based knowledge engineering and enterprise modelling. Research report. Institute AIFB. University of Karlsruhe (TH). Germany, 1996.

[Eriksson et al. 95] H. Eriksson, Y. Shahar, S.W. Tu, A.R. Puerta, and M. Musen: Task Modeling with reusable problem-solving methods. *Artificial Intelligence*, 79 (2) 1995, pp.293-326.

[Fensel et al. 97] D. Fensel, J. Angele, and R. Studer: The Knowledge Acquisition and Representation Language KARL. *IEEE Transactions on Knowledge and Data Engineering,* to appear.

[Fensel 95] D. Fensel: *The Knowledge Acquisition and Representation Language KARL.* Kluwer Academic Publisher, Boston, 1995.

[Floyd 84] C. Floyd: A Systematic Look at Prototyping. In: R. Budde et al. (eds.), *Approaches to Prototyping*, Springer-Verlag, Berlin, 1984, 1-18.

[de Hoog et al. 94] R. de Hoog, B. Benus, C. Metselaar, M. Vogler, and W. Menezes: Organisation Model: Model Definition Document. ESPRIT Project P5248 KADS-II, report KADS-II/M6/UvA/041/3.0, University of Amsterdam, 1994.

[Kangassalo et al. 95] H. Kangassalo, H. Jaakkola, S. Ohsuga, and B. Wangler (eds): *Information Modelling and Knowledge Bases IV*, IOS Press, 1995.

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

[Kirikova and Bubenko 94] 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.

[Klinker et al. 93] G. Klinker, D. Marques, and J. McDermott: The Active Glossary: Taking Integration Seriously. *Knowledge Acquisition*, **5**, 173-197.

[Landes and Studer 95] D. Landes and R. Studer: The treatment of non-functional requirements in MIKE. In: *Proceedings of the 5th European Software Engineering Conference (ESEC'95).* Barcelona, Spain, September 1995. LNCS 989, Springer, Berlin, 1995.

[Landes 94] 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.

[Neubert 93] S. Neubert: Model Construction in MIKE (Model Based and Incremental Knowledge Engineering). In: *Current Trends in Knowledge Acquisition - EKAW'93, 7th European Knowledge Acquisition Workshop*, Toulouse, France, LNAI 723, Springer, Berlin, 1993.

[Oberweis et al. 94] A. Oberweis, G. Scherrer, W. Stucky: INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems. *Information Systems*, Vol. 19, No. 8, 1994, pp.643-660.

[Österle 95] H. Österle: *Business in the Information Age. Heading for New Processes*. Springer, Berlin,1995.

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

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

[Scheer 92] A.-W. Scheer: *Architecture of Integrated Information Systems, Foundations of Enterprise Modelling.* Springer, Berlin , 1992.

[Scheer 94] A.-W. Scheer: *Business Process Engineering: Reference Models for Industrial Enterprises.* Springer, Berlin, 2nd ed. 1994.

[Schreiber et al. 93] 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.

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

[Schreiber and Wielinga 93] G. Schreiber and B. Wielinga: KADS and Conventional Software engineering. In: [Schreiber et al. 93], pp. 151-166.

[Steels et al. 94] L. Steels, G. Schreiber, and W. van de Velde (eds.): *A Future for Knowledge Acquisition. Proceedings of the 8th EKAW 94*. LNAI 867. Springer, Berlin 1994.

[Vanwelkenhuysen and Mizoguchi 94] 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.

[Yost et al. 94] G.R. Yost, G. Klinker, M. Linster, D. Marques, and J. McDermott: The SBF Framework, 1989-1994: From Applications to Workplaces. In: [Steels et al. 94], pp.318-339.