Viewpoint-based Meta Model Engineering

Stephan Kurpjuweit, Robert Winter

Institute of Information Management
University of St. Gallen
Mueller-Friedberg-Strasse 8
9000 St. Gallen, Switzerland
stephan.kurpjuweit@unsig.ch
robert.winter@unisg.ch

Abstract: Work systems are complex artifacts that address the concerns of a large and diverse group of stakeholders. These concerns must be reflected in the models which are created as used in the development process. Current work systems engineering methods assume that concerns are more or less mutually independent and can be addressed sequentially. We argue - in analogy to other engineering disciplines - that this assumption is too restrictive. To facilitate the creation of models that simultaneously express multiple stakeholder concerns, we propose an approach which systematically elicits the stakeholder concerns, and derive a customized meta model from these concerns. We also show how this approach has been applied in an industrial case study, and propose a set of extensions to the method engineering meta model that allow method engineers to include stakeholder concerns in work system design methods.

1 Introduction

Models play a pivotal role in information systems and work systems[1] engineering: Among other purposes, models of the system under construction serve as a blueprint for its implementation, to reason about its prospective properties, to structure its development process, to decompose the system into mutually independent sub problems and to communicate it among the various stakeholders in the development process.

Like almost any engineered artifact, work systems are inherently complex and must address the concerns of a large and diverse group of stakeholders. These include participants in the design and implementation process of the work system as well as stakeholders concerned with the properties of the work system to be implemented. The models of the work system created throughout its development process must adequately

[1] Following the argumentation of Alter, we prefer the term *work system (WS)* over the more specific term *information system (IS)*. A work system is defined as "a system in which human participants and/or machines perform work using information, technology, and other resources to produce products and/or services for internal or external customers" [Al06b].. Information systems can thus be seen as a specific subtype of work systems [Al03], [Al06a].. Therefore, throughout this paper we refer to the results of the design process as work systems.

reflect the concerns of these various stakeholders. The applied modeling concepts must appropriately express these concerns.

Most approaches in information systems and work systems engineering put concerns on the same level with process phases and the artifacts created within these phases. This reflects the assumption that concerns are more or less mutually independent and can thus be addressed one by one in sequential order (e.g. [FS95, Sc01, Wi03]). Sutton and Rouvellou [SR01] argue that this view is too restrictive because most concerns cut across process phases and the corresponding artifact types. Although this observation has been made for the domain of software engineering, it seems to be reasonable to assume that it holds true for the even broader domain of work systems engineering.

Meta models define the modeling concepts that can be used to describe models [KLC05]. Meta models can thus be seen as models of modeling languages [Fa05] and "the task of creating a meta model is the task of creating a language that is capable to describe the relevant aspects of a subject under consideration that are of interest for the future users of the created models" [Hö07].

To summarize: Innovative engineering approaches will address increasingly complex artifacts (work systems instead of information systems) by means of models that simultaneously express multiple crosscutting stakeholder concerns. Consequently, also the applied modeling concepts and meta models will be more complex and should be constructed systematically. Though a large theoretical foundation is available in the area of conceptual modeling and language construction (e.g. [BP06, LSS94, Mo05, STW03, We03, WW02]), only little work has been done to address the systematic construction of meta models that explicitly and comprehensibly represent the concerns of the various stakeholders.

In this paper we propose a systematic and applicable approach to elicit the concerns and the information needs from all stakeholders of a work system and to derive a customized meta model from these concerns and needs. Our approach incorporates and complements existing approaches and insights from the available theoretical body of knowledge. It has been applied and iteratively refined in case studies with industry partners.

The paper is structured as follows: Sections 2 and 3 discuss key concepts that lead to requirements or solution ideas for our approach. Section 4 summarizes the requirements for viewpoint-based work systems engineering. Section 5 presents our approach to meta model engineering. Section 6 discusses how our approach can be integrated into the method engineering meta model, and section 7 briefly describes an industrial application of the proposed approach.

2 Models, Meta Models, Stakeholders, Concerns, and Viewpoints

According to Stachowiak [St73], a model possesses three essential properties: the representation property (a model represents an original, e.g. the work system under consideration), the reduction property (a model represents a relevant subset of all possible properties of the original), and the pragmatic property (a model serves a purpose). Although many possible modeling purposes have been discussed, three main categories can be identified (cf. [LHM95]):

(1) Documentation and communication (here: to document the work system as-is and to communicate it among the stakeholders)
(2) Analysis and explanation (here: to analyze how the work system performs with respect to certain concerns and to identify strategies how it may be improved)
(3) Design (here: to prescribe a to-be blueprint of the work system).

A model is created by a modeler and interpreted by one or more users with respect to a certain purpose [Le04]. As modelers and users of a model are not necessarily identical, it is important to ensure that both parties are able to understand the model.

Models conform[2] to meta models. Meta models define the modeling concepts that can be used to describe models [KLC05]. A meta model is thus a model of a modeling language [Fa05]. As a meta model itself is a model, it may conform to a meta meta model. Though in this way a hierarchy of models and meta models can be carried to the n^{th} level, in practice the definition of the meta meta model is usually reflexive [Hö07].

According to Harel and Rumpe [HR00], a modeling language has syntax (defining the notational aspects) and semantics (defining the meaning). Additionally Kühn introduces the notation as explicit representation of the language elements [Kü04]. In this view a meta model defines the abstract syntax of a modeling language (i.e. the modeling constructs and valid ways to combine them [Hö07]), while the notation defines the concrete syntax [Di03].

As mentioned before, work systems are inherently complex and must address the concerns of a large and diverse group of stakeholders. These include systems architects, project managers, sponsors, implementers, and change agents who are participants of the design and implementation process, as well as customers, employees, managers, system operators, outsourcing partners, or the workers' council which are stakeholders concerned with the properties of the implemented work system[3]. Catalogs of – mostly technical – concerns have been published for software and information systems engineering (cf. [Al00, Ba04, CE00]). These include quality concerns like security (cf. [CE00]) or system performance (cf. [Al00]) as well as design related concerns like the structure and representation of data (cf. [CE00]). In the context of work systems engineering, also strategic and organizational concerns like business service realization and business process efficiency should be considered (cf. [Do04]). Based on the definition suggested by Sutton and Rouvellou [SR01] we define a concern as a matter of interest in a work system. Accordingly, a stakeholder is defined as a person or an organization that has a concern in a work system.

---

[2] We agree with Bézivin [Be05]. and Favre [Fa05] who argue that the term *conforms to* should be preferred over *instance of*.

[3] This distinction is similar to the distinction between design time and run time concerns of a software system.

The models of the work system must adequately reflect the concerns of the various stakeholders. The stakeholders' concerns and needs impact models of the work system in two ways: First, syntax and notation of the modeling language must be appropriate for the stakeholders' educational background (internal quality). This is for example relevant if employees with a business background must be able to interpret a procedural model of the work system.

Second, the design of the work system itself (i.e. the model content) must address the requirements of stakeholders to ensure that the work systems implemented on the basis of the models satisfies their requirements (external quality). This is for example the case if stakeholders responsible for the security of a work system need to ensure that appropriate technical mechanism (e.g. firewalls, encrypted network connections) or appropriate organizational mechanisms (e.g. policies to have transactions reviewed by a second set of eyes) are in place. In the latter case, the modeling language must also provide appropriate modeling constructs to express the design decisions made to address the stakeholders' concerns. The distinction between internal and external quality originates from the ISO/IEC 9126 standard [ISO01] and has been adopted to evaluate the quality of conceptual models (cf. [Mo05]).

In software engineering and requirements engineering the concept of viewpoints has been discussed since the early 1990s (cf. [Fi92, KS92, Nu94]) to simultaneously consider multiple concerns in system description and design [Do04]. The IEEE-1471 standard for architecture description [IEE00] contains the most prominent conception of viewpoints. Despite all differences between the various notions of viewpoints that have been published, most authors agree that a viewpoint describes appropriate modeling machinery (e.g., a modeling language and/or a modeling method) to capture one or more related concerns about a system. The viewpoint definition most suited for the purpose of this paper has been given by Bayer [Ba04]: "A viewpoint covers a number of concerns and defines the information associated with the concern in the metamodel." In our approach viewpoints are a major concept to structure the stakeholders' requirements and to derive meta model fragments which satisfy these requirements.

3 Methods and Situational Method Engineering

The generic structure of development processes is codified in methods. Brinkkemper defines a method as "[…] an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development product" [Br96]. The discipline of method engineering (ME) is concerned with the design, the construction, and the adaption of methods. Though most method concepts discussed in the ME discipline aim at engineering and transforming information systems [TA03], the concept of a method can also be applied at the scope of work systems [Ba05].

Based on a review of approaches to method implementation and method construction, Heym [He93] and Gutzwiller [Gu94] identified five constituent elements of methods: design activities, documents specifying design results, roles, techniques, and the meta model of the method. Braun et al. [Br05] validated this set of concepts for the description of generic methods by analyzing twelve scientific publications in the domain of method engineering . Thus, these five elements of work system design methods can be seen as a "core" method engineering meta model (unshaded entities in figure 4). As indicated by the method engineering meta model, design results conform to the meta model, which is an integral component of a method.

Recent research in method engineering has stressed the fact that methods are generic artifacts and must be adapted to the specific project type and context factors at hand [Br96, Ha97]. [Bu07] differentiates between the "project type" (defined in terms of the initial situation and the project goals to be achieved) and the "context" (defined in terms of environmental factors that may impact the project execution) and proposes an extension to the core method engineering meta model by adding the concepts "context", and "project type". A "situation" is a combination of certain contexts and certain project types. A "method fragment" is a generalized concept for design activities and techniques and can be adapted to a specific situation by means of "adaption mechanisms". Figure 4 includes these extension concepts as entities shaded in dark grey.

As entire work systems are rarely designed from scratch, methods are usually applied in transformation projects (cf. [Bu07]). This raises the need to differentiate various states of the work system (e.g., as-is vs. to-be). It should be noted that as-is and to-be models of the work system may not only differ regarding represented content, but also regarding modeling concepts applied - and thus regarding the underlying meta model. This is typically the case if the transformation project applies a new paradigm to structure the work system. E.g., new paradigms are applied when moving from an application-oriented IT landscape to a service-oriented IT landscape, or from a hierarchical organizational structure to a matrix-oriented organizational structure.

4 Goals for an Approach to Meta Model Engineering

From the description of key concepts in sections 2 and 3, the following key requirement categories for systematical, viewpoint-based work systems engineering can be derived:

(1) The fact that meta models are constructed as integral parts of work system design methods must be reflected.
(2) The concerns and information needs of both participants in the work system construction process as well as stakeholders concerned with the properties of the work system must be considered.
(3) The approach must be independent of a specific meta meta model and thus independent of a specific modeling technique (cf. section 5).
(4) As an engineering approach it should facilitate reuse of meta models, provide a design rationale for modeling decisions, and address the need to adapt meta models to specific project types and to specific context factors.

5 A Systematic Approach to Meta Model Engineering

The purpose of our approach is to systematically elicit the concerns and the information needs from all stakeholders of a work system and to derive a meta model from these concerns and needs. Since our approach intends to be independent of specific meta modeling methodologies and meta meta models (cf. section 4, goal 3), we construct our engineering approach around any epistemologically valid meta modeling technique[4] that supports modeling and integration of meta models. Researchers have proposed meta modeling techniques based on the ER Model [NKF93, STM88], Attribute Grammars [Ka89, So95], Predicate Logic [NKF93], the object-oriented modeling approach [Ju00, Kü03] or other approaches [Aj96] (cf. [BSH99]).

The basic idea of our approach is to partition the complete set of modeling requirements into viewpoints, to develop a meta model fragment for each viewpoint, and to integrate the meta model fragments into a comprehensive meta model. By that means, our approach implements three engineering principles: First, the complex modeling problem is partitioned into less complex and mutually independent sub problems (divide and conquer). Second, the meta model fragments describe encapsulated solutions, which can be reused in similar situations. Third, the purpose of meta model elements can be traced back to the modeling requirements.

Consequently, the proposed approach consists of three main steps: "Requirements Elicitation", "Meta Model Fragment Selection or Design", and "Meta Model Fragment Integration". We add two auxiliary steps:" Identification of Relevant Concerns" (to gain an overview of all concerns to be addressed) and "Viewpoint Relationship Overview" (to check the set of viewpoint for completeness and consistency).

[4] Similarly to software engineering methods that are typically independent of specific programming languages.

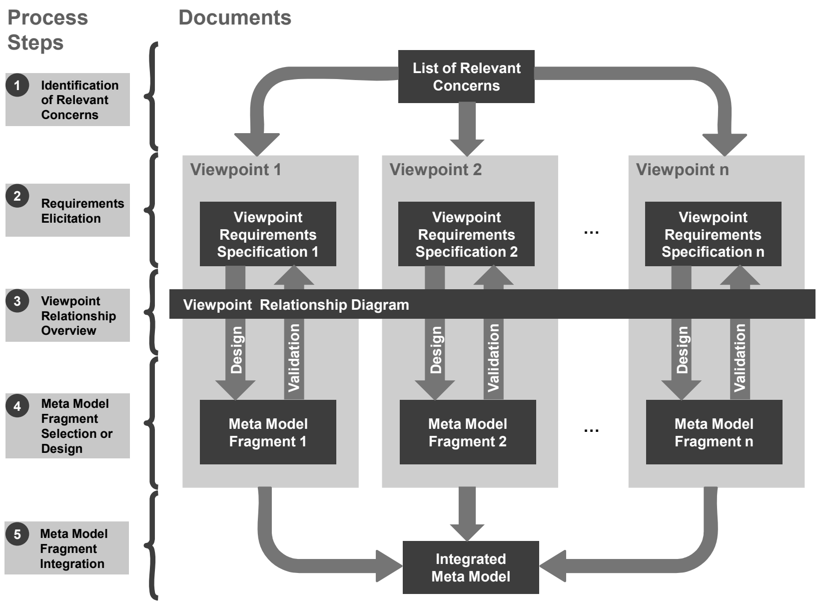


Figure 1: Viewpoint-based Meta Model Engineering – Overview

Figure 1 illustrates the process steps and documents of the proposed approach. Steps 2 and 4 are performed for each viewpoint. Thus, in a modeling project the steps can either be performed in sequential order if the scope of the meta model or the criteria to partition the requirements in viewpoints are unclear at the beginning, or the steps can be performed iteratively for each viewpoint. Our approach addresses two general application scenarios:

(1) The initial definition of reusable viewpoint as part of a new method development project: In this case the desired degree of generality (i.e. the project types and the context factors for which the viewpoints should be applicable) must be considered.
(2) The application of viewpoints in a concrete project: In this case the selection and integration of existing meta model fragments are the main activities.

The modeling projects we conducted with industry partners revealed aspects of both scenarios: For some concerns it was possible to reuse pre-defined meta model fragments, while for other concerns new meta model fragments had to be created.

The five steps of the proposed approach are specified in detail as follows:

Step 1: Identification of Relevant Concerns

The goal of step 1 is to assemble a broad list of relevant concerns form a large and diverse group of stakeholders. A reference list of concerns can be used as a starting point (e.g. [Al00, CE00]). The set of concerns however should be specific for the project type and the context factors at hand. Ideally this activity is performed within a workshop where all important stakeholders are present (or at least represented).

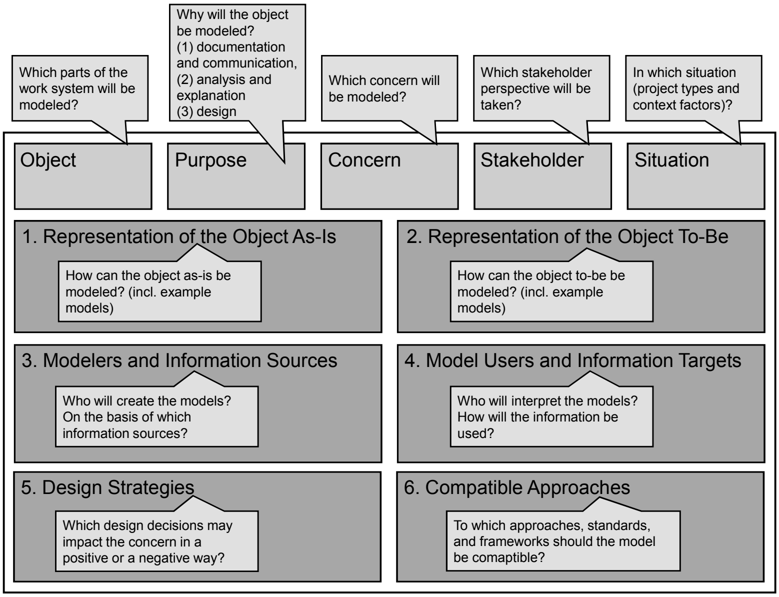


Figure 2: Viewpoint Requirements Template (VRT)

Step 2: Requirements Elicitation

The technique applied in this step is adapted from the Goal-Question-Metric (GQM) Method [SB99], an approach to systematically derive situational metrics from questions that are in turn derived from stakeholder goals: In structured interviews with the individual stakeholders, each concern identified in step 1 is refined on the basis of the viewpoint requirements template (VRT) shown in figure 2 (a tailored version of the goal template provided by the GQM method). Related concerns may be refined together in one VRT. The VRT consists of a viewpoint goal and six additional elements to be filled in by the stakeholder. The viewpoint goal can be paraphrased: “Represent the OBJECT to {*document and communicate* or *analyze and explain* or *design*} the CONCERN from the perspective of STAKEHOLDER in SITUATION.” (cf. [SB99])

Next, specific questions are derived on the basis of the viewpoint goal: What questions should the model answer to achieve the viewpoint goal? These questions represent requirements regarding the information content of the models and thus the modeling concepts included in the meta model fragment.

Step 3: Viewpoint Relationship Overview

This step is optional but may be useful to gain a model-centric overview of the method under construction and to check the information gathered in the different VRTs for correctness, completeness, and consistency. One overall viewpoint relationship diagram is created that summarizes the information about the relationships of the different viewpoints; Figure 3 illustrates the available vocabulary: "model/document type", "modeler/model user", "stakeholder", "information source/target" as well as the relationships "manual transformation", "automated transformation" (each either between two model/document types or a document/model type and an information source/target), and "association" (between stakeholder and model/document type, modeler/model user and model/document type). Figure 6 shows an example of a model relationship diagram.

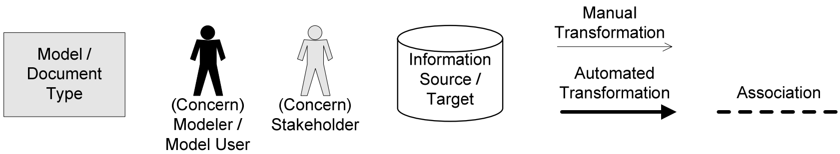


Figure 3: Viewpoint Relationship Diagram (Legend)

Step 4: Meta Model Fragment Selection or Design

In this step the individual viewpoint specifications are complemented by adding an appropriate meta model fragment. The meta model fragment can either be designed from scratch or selected from a viewpoint catalogue. The meta model fragment must be validated against its requirements by answering the questions derived in step 2.

To design the meta model fragment from scratch, an appropriate modeling technique should be applied. As our approach intends to be independent of specific modeling techniques and meta meta models (see above), we only specify the following constraints:

(1) The meta model must be minimal, i.e. only contain elements that are motivated by the information needs specified in the viewpoint. Otherwise effort may be spent later on to model content that cannot be interpreted with respect to a viewpoint goal (cf. [SB99]).
(2) A design rationale for the individual meta model elements must be recorded (to address goal 4 as stated in section 4).
(3) The semantics of the meta model elements must be clarified at least intuitively to avoid misunderstandings between different stakeholder groups.

A simple and straightforward way to achieve this is a table that provides for each meta model element a short rationale, example instances, and if necessary also negative examples that shall not be modeled as instances of the meta model element.

Step 5: Meta Model Fragment Integration

Once meta model fragments for all relevant viewpoints are available, these fragments must be integrated into one integrated meta model that expresses the interrelationships between the various viewpoints. Again, the concrete approach for meta model integration depends on the underlying meta meta model. Researchers have published several approaches to meta model integration (e.g. [BSH99, ES06, Kü03, RR01]). In general, the following issues must be addressed: (a) Terminology must be adjusted to ensure that semantically similar concepts have the same name and that semantically different concepts have different names (cf. [ES06, RR01]). (b) Generalizations must be created if two concepts have similar semantics but different structures (cf. [RR01]). (c) Specializations must be created if one concept is a specialization of another concept (cf. [RR01]). (d) If the same information content is represented in different ways, such redundancies need to be removed (cf. [RR01]). (e) In order to relate meta model fragments, interface modeling concepts may have to be introduced (cf. [ES06, Kü03]).

In order to ensure that all concerns and information needs are covered, the integrated meta model should also be validated against the questions noted in the individual viewpoint specifications and against the viewpoint relationship diagram.

6 Extensions to the Method Engineering Meta Model

The extended method engineering meta model proposed in [Bu07] (cf. section 3) does not reflect the viewpoint-based design of meta model fragments as presented in the paper at hand: While design activities and techniques are considered as method fragments that constitute the building blocks of methods and that can be configured and composed according to the requirements of specific project types and context characteristics, the meta model is still treated as a monolithic artifact.

To reflect the viewpoint-based meta model engineering approach discussed in the previous sections, we propose another set of extensions to the method engineering meta model by adding the following concepts:

- "stakeholder" who has one or more concerns and who might hold one or more roles in the development project.
- The stakeholder's concerns are addressed by "design strategies" which are applied in the design activities.
- A "meta model fragment" provides the modeling concepts to capture the design decisions which are based on a design strategy in order to address a concern. Thus, meta model fragments are concern-related. The complete meta model of a method is an integration of all relevant meta model fragments as described in section 5.
- As defined in section 2, "viewpoints" package one or more concerns together with related meta model fragments.
- The concept "notation" introduces the differentiation between the abstract and the concrete syntax as proposed by Kühn [Kü04] (cf. section 2).

Figure 4 illustrates the proposed, extended method engineering meta model. The additions specified in this section are shaded in light grey. The proposed meta model extensions are in accordance with existing viewpoint-based approaches from software and requirements engineering (cf. section 2) and reflects the ideas presented in the paper at hand.

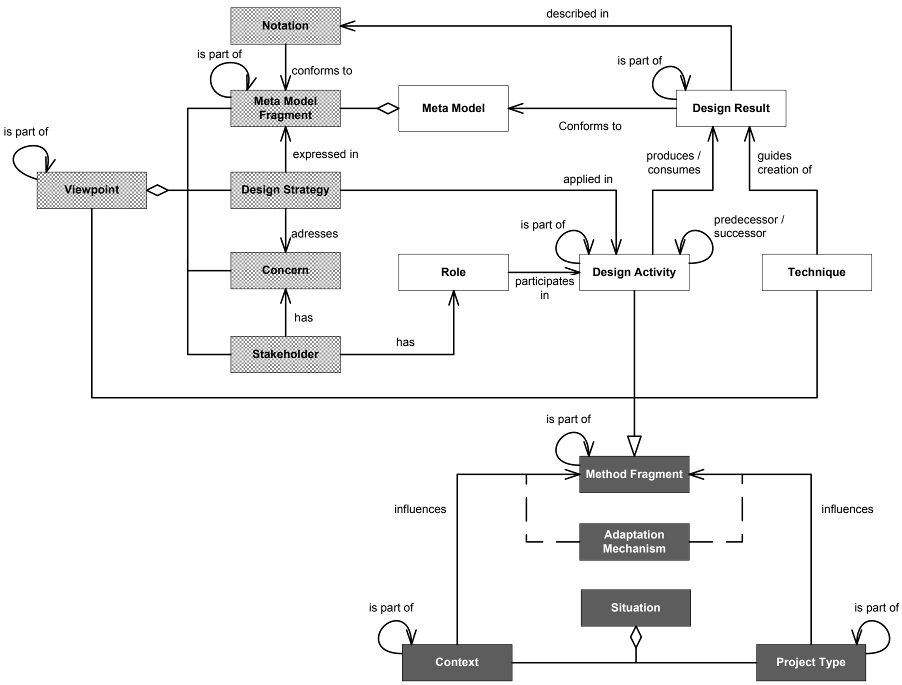


Figure 4: Extended Method Engineering Meta Model (Notation adapted from UML class diagram)

7 Case Study: Modeling IT Architectures

In this section we briefly describe an industrial case study in which our approach has been applied. The meta modeling project was conducted with a mid size financial service provider in Germany. The goal was to establish a meta model that facilitates management and planning of the organization´s IT architecture. In the requirements elicitation phase of the project 14 stakeholder groups were interviewed in workshops resulting in a total of 45 essential requirement statements which could be structured in 16 viewpoint specifications.

The meta model fragments derived for the individual viewpoints were modeled from scratch using the object-oriented modeling approach [Ju00, Kü03] and a variant of the UML class diagrams as notation. Figure 5 shows the integrated meta model. In the meta model cardinalities and identifiers of relationships as well as attributes of classes are omitted.

As indicated by the method engineering meta model, the concept of a viewpoint is also a method fragment and can thus be adapted to specific situations. This is necessary because the concerns of a stakeholder and the design strategies to address these concerns may depend on context factors and project types. For example, the concerns of the workers' council will depend on the size and the economic situation of the enterprise. Another example is that available design strategies to optimize the alignment between business processes and available IT functionalities will be different in context of a standard software as opposed to a best of breed IT strategy. Considering viewpoints as method fragments as well as composing meta models from concern-related meta model fragments are aids to overcome the monolithic approach to meta modeling which dominates traditional method engineering.

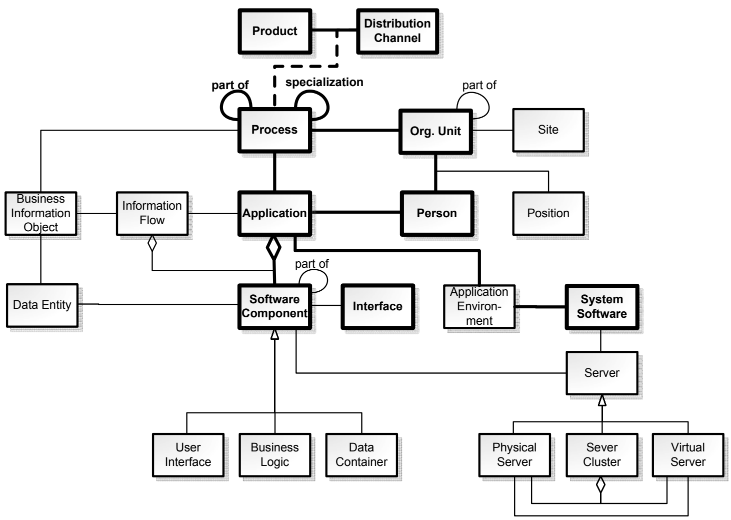


Figure 5: Complete Meta Model (simplified)

The proposed approach can be seen as a meta method, i.e. a method to engineer meta models as part of methods. In this light, the method engineering meta model presented in figure 4 becomes the meta model of our meta method, and the design results incorporated in the meta method conform to this meta model.

The appendix contains further case study material that exemplifies intermediate documents which were created throughout the meta modeling project.

8 Conclusion and Further Research

In our paper we presented a new approach to meta model engineering: By means of a five step process, the modeling requirements from all stakeholders of a work system are elicited, specified as viewpoints, and refined into meta model fragments which are in turn integrated into a comprehensive meta model. In this way meta models are constructed that simultaneously reflect the concerns of multiple stakeholders. Such meta models will be an important component of innovative work system design methods. Furthermore, we proposed extensions to the method engineering meta model that allow the method engineer to include stakeholder concerns in descriptions of work system design methods.

In an industrial case study our approach turned out to be useful to construct meta models which address multiple stakeholder concerns. Further research should focus on an evaluation of the proposed approach as part of the design research process. For such an evaluation we will conduct further case studies (to evaluate that the concerns of various stakeholders can be elicited and reflected properly) as well as experiments (to evaluate that the integration of meta model fragments and the verification of the integrated meta model lead to reproducible results). Based on the initial contribution presented in this paper, there are three broad directions to extend this work: First of all, a handbook of viewpoints for work system design can be assembled on the basis of existing research results from concern-focused research communities and on the basis of further industrial case studies. Second, the mechanisms to adapt viewpoints to specific project types and context factors can be formalized. This could for example be achieved by incorporating approaches from reference modeling (e.g. [Be02, Br03]). Third, our approach could be extended to provide concrete guidelines for the design and integration of meta model fragments on the basis of specific meta meta models. This requires the evaluation and integration of existing meta modeling techniques (e.g. [NKF93, STM88]).

References

[Aj96] *Ajisaka, Tsuneo*: The software quark model: a universal model for CASE repositories. In: Information and Software Technology 38 (1996), S. 173-180.

[Al00] *Aldrich, J.*: Challenge Problems for Separation of Concerns. OOPSLA 2000 Workshop on Advanced Separation of Concerns, Minneapolis, USA, 2000.

[Al03] *Alter, Steven*: 18 Reasons Why IT-reliant Work Systems Should Replace “The IT Artifact” as the Core Subject Matter of the IS Field. In: Communications of the Association for Information Systems 12 (2003) 23, S. 366-395.

[Al06a] *Alter, Steven*: Work Systems and IT Artifacts - Does the Definition Matter? In: Communications of the Association for Information Systems 17 (2006) 14, S. 299-313.

[Al06b] *Alter, Steven*: The Work System Method. 1, Work System Press, Larkspur, CA 2006.

[Ba04] *Bayer, Joachim*: View-based Software Documentation. Ph.D. Thesis, Universität Kaiserslautern, Kaiserslautern, 2004.

[Ba05] *Baumoel, Ulrike*: Strategic Agility through Situational Method Construction. In: *Reichwald, Ralf; Huff, Anne Sigismund (Hrsg.)*: Proceedings of the European Academy of Management Annual Conference (EURAM2005). http://www.euram2005.de, 2005,
[Be02] *Becker, Joerg; Algermissen, Lars; Delfmann, Patrick; Knackstedt, Ralf*: Referenzmodellierung. In: Das Wirtschaftsstudium 30 (2002) 11, S. 1294-1298.
[Be05] *Bézivin, J.*: On the Unification Power of Models. In: Software and System Modeling 4 (2005) 2, S. 171-188.
[BP06] *Becker, Jörg; Pfeiffer, Daniel*: Konzeptionelle Modellierung – ein wissenschaftstheoretischer Forschungsleitfaden. In: Lehner, Franz; Nösekabel, Holger; Kleinschmidt, Peter (Hrsg.): Multikonferenz Wirtschaftsinformatik (MKWI 2006), Band 2, 2006.
[Br03] *vom Brocke, J.*: Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen. Ph.D. Thesis, Logos, Berlin, 2003.
[Br05] *Braun, Christian; Wortmann, Felix; Hafner, Martin; Winter, Robert*: Method Construction - A Core Approach to Organizational Engineering. In: *Haddad, Hisham; Liebrock, Lorie M.; Omicini, Andrea; Wainwright, Roger L. (Hrsg.)*: Proceedings of the 20th Annual ACM Symposium on Applied Computing (SAC 2005). ACM, Santa Fe 2005, 1295-1299.
[Br96] *Brinkkemper, Sjaak*: Method Engineering - Engineering of Information Systems Development Methods and Tools. In: Information and Software Technology 38 (1996), S. 275-280.
[BSH99] *Brinkkemper, Sjaak; Saeki, Motoshi; Harmsen, Anton Frank*: Meta-Modelling Based Assembly Techniques for Situational Method Engineering. In: Information Systems 24 (1999) 3, S. 209-228.
[Bu07] *Bucher, Tobias; Klesse, Mario; Kurpjuweit, Stephan; Winter, Robert*: Situational Method Engineering - On the Differentiation of "Context" and "Project Type". IFIP WG8.1 Working Conference on Situational Method Engineering (ME07), 2007.
[CE00] *Czarnecki, K.; EIsenecker, U.*: Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000.
[Di03] *Dijkman, Remco M.; Quartel, DIck A.C.; Pires, Luís Ferreira; Svan Sinderern, Marten J.*: An Approach to Relate Viewpoints and Modeling Languages. Seventh IEEE International Enterprise Distributed Object Computing Conference, 2003.
[ES06] *Emerson, Matthew; Sztipanovits, Janos*: Techniques for Metamodel Composition. The 6th OOPSLA Workshop on Domain-Specific Modeling, OOPSLA, 2006.
[Fa05] *Favre, Jean-Marie*: Foundations of Meta-Pyramids: Languages vs. Metamodels - Episode II: Story of Thotus the Baboon. In: *Bézivin, J.; Heckel, R. (Hrsg.)*: Language Engineering for Model-Driven Software Development. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Dagstuhl 2005,
[Fi92] *Finkelstein, A.; Kramer, J.; Nuseibeh, B.; Finkelstein, L.; Goedicke, M.*: Viewpoints: a framework for integrating multiple perspectives in system development. In: International Journal on Software Engineering and Knowledge Engineering 2 (1992) 1, S. 31-58.
[FS95] *Ferstl, Otto K.; Sinz, Elmar J.*: Der Ansatz des Semantsichen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. In: Wirtschaftsinformatik 37 (1995) 3, S. 209-220.
[Gu94] *Gutzwiller, Thomas*: Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. Physica, Heidelberg 1994.
[Ha97] *Harmsen, Frank*: Situational Method Engineering. Moret Ernst & Young Management Consultants, Utrecht 1997.
[He93] *Heym, Michael*: Methoden-Engineering - Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme. Ph.D. Thesis, University of St. Gallen, 1993.

[Hö07] *Höfferer, Peter*: Achieving Business Process model Interoperability Using Metamodels and Ontologies. In: Österle, Hubert; Schelp, Joachim; Winter, Robert (Hrsg.): 15th European Conference on Information Systems (ECIS 2007), 2007.

[ISO01] *ISO/IEC*: Standard 9126: Software Product Quality. International Standards Organization (ISO, International Electrotechnical Commission (IEC), 2001.

[Ju00] *Junginger, Stefan; Kühn, Harald; Strobl, Robert; Karagiannis, Dimitris*: Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation. In: Wirtschaftsinformatik 42 (2000) 5, S. 392-401.

[Ka89] *Katayama, Takuya*: A Hierarchical and Functional Software Process Description and its Enaction. 11th Int. Conf. on Software Engineering, 1989.

[KLC05] *Klint, Paul; Lämmel, Ralf; Verhoef, Chris*: Towards an engineering discipline for grammarware. In: ACM Transactions on Software Engineering and Methodology (TOSEM) 14 (2005) 3, S. 331-380.

[KS92] *Kontonya, G.; Sommerville, I.*: Viewpoints for requirements definition. In: IEE/BCS Software Engineereing Journal 7 (1992) 6, S. 375-387.

[Kü03] *Kühn, Harald; Bayer, Franz; Junginger, Stefan; Karagiannis, Dimitris*: Enterprise Model Integration. In: Bauknecht, K.; Tjoa, A. M.; Quirchmayer, G. (Hrsg.): Fourth International Conference EC-Web 2003 – Dexa 2003, Berlin et al., 2003.

[Kü04] *Kühn, Harald*: Methodenintegation im Business Engineering. Ph.D. Thesis, Universität, Wien, 2004.

[Le04] *Leist, Susanne*: Methoden zur Unternehmensmodellierung - Vergleich, Anwendungen und Diskussionen der Integrationspotenziale. Habilitation, Institut für Wirtschaftsinformatik, Univeristät St. Gallen, St. Gallen, 2004.

[LHM95] *Lehner, Franz; Maier, Ronald; Hildebrand, Knut*: Wirtschaftsinformatik: Theoretische Grundlagen. 1, Carl Hanser Verlag, München, Wien 1995.

[LSS94] *Lindland, Odd Ivar; Sindre, Guttorm; Solvberg, Arne*: Understanding Quality in Conceptual Modeling. In: IEEE Software 11 (1994) 2, S. 42-49.

[Mo05] *Moody, Daniel L.*: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. In: Data & Knowledge Engineering 55 (2005) 3, http://dx.doi.org/10.1016/j.datak.2004.12.005, S. 243-276.

[NKF93] *Nuseibeh, B.; Kramer, J.; Finkelstein, A.*: Expressing the relationship between multiple view in requirements specification. 15th Int. Conf. on Software Engineering, 1993.

[Nu94] *Nuseibeh, B.*: A Multi-Perspective Framework for Method Integration. Ph.D. Thesis, University of London, London, 1994.

[RR01] *Ralyté, Jolita; Rolland, Colette*: An Assembly Process Model for Method Engineering. In: Dittrich, Klaus R.; Geppert, Andreas; Norrie, Moira C. (Hrsg.): Advanced Information Systems Engineering, Berlin, 2001.

[SB99] *van Solingen, Rini; Berghout, Egon*: The Goal/Question/Metric Method. 1, McGraw Hill, London 1999.

[Sc01] *Scheer, August-Wilhelm*: ARIS – Modellierungsmethoden, Metamodelle, Anwendungen. 4, Springer, Berlin Heidelberg 2001.

[So95] *Song, X.*: A framework for understanding the integration of design methodologies. In: ACM SIGSOFT Software Engineering Notes 20 (1995) 1, S. 46-54.

[SR01] *Sutton, S. M.; Rouvellou, I.*: Issues in the Design and Implementation of a Concern-Space Modeling Schema. Advanced Separation of Concerns Workshop, Toronto, Canada, 2001.

[St73] *Stachowiak, H.*: Allgemeine Modelltheorie. Springer, New York, Wien 1973.

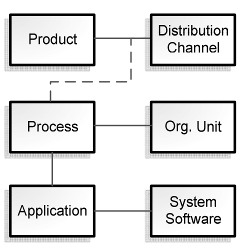
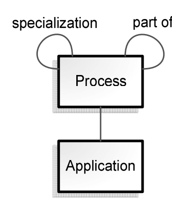
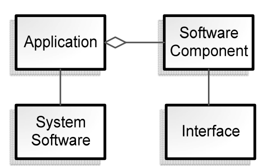
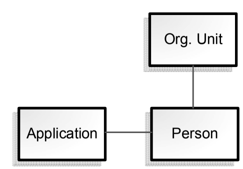
[STM88] *Sorenson, Paul G.; Tremblay, Jean-Paul; McAllister, Anderew J.*: The Metaview System for Many Specification Environments. In: IEEE Software 5 (1988) 2, S. 30-38.

[STW03] *Shanks, Graeme; Tansley, Elizabeth; Weber, Ron*: Using Ontology To Validate Conceptual Models. In: Communications of the ACM 46 (2003) 10, S. 85-89.

[We03] *Weber, Ron*: Conceptual Modelling and Ontology: Possibilities and Pitfalls. In: Journal of Database Management 14 (2003) 3, S. 1-20.
[Wi03] *Winter, Robert*: Modelle, Techniken und Werkzeuge im Business Engineering. In: *Österle, Hubert; Winter, Robert (Hrsg.)*: Business Engineering - Auf dem Weg zum Unternehmen des Informationszeitalters. Springer, Berlin etc. 2003, 87-118.
[WW02] *Wand, Yair; Weber, Ron*: Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. In: Information Systems Research 13 (2002) 4, S. 363-376.

Appendix: Case Study Material

Table 1: Viewpoint Refinement (simplified)

Viewpoint	IT Consolidation	Business IT Alignment	Component Reuse	Ownership
Object	Processes, Applications	Processes, Applications	Software architecture	IT-related artifacts
Purpose	Analysis	Analysis	Design	Documentation
Concern	Cost of application operations and maintenance	Providing adequate IT for business processes	Cost of application development	Correct implementation of ownership policies
Sakeholder	Application architect	Process owner	Software architect	IT audit
Design Strategies	Consolidation of applications that are in use for a similar purposes / Consolidating of system software of the same type (e.g., DBMS, WFMS)	Providing IT functionalities for each process step / Reduction of media breaks	Reuse of software components across multiple applications / Reuse of system software (e.g., DBMS, WFMS)	Assigning explicit owners to applications and other IT-related artifacts (like information objects, components, environments, etc.)
Questions	Which applications are used in the individual processes (sorted by organizational unit, product, distribution channel)? / Which system software of the same type is currently in use?	Which process activities are not IT supported? / Which processes include media breaks? / Which activities are supported by multiple applications?	Which components are available in existing applications? / Which interfaces are available to use these components? Which system software of the different types is currently in use?	Are there applications for which no owners have been defined? Are there applications that have not been audited for more than two years?
Meta Model Fragment				

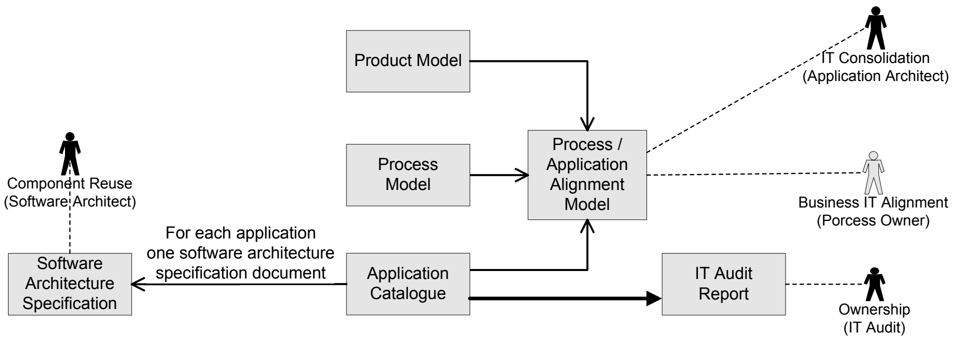


Figure 6: Viewpoint Relationship Diagram (simplified)

Table 1 illustrates the refinement of four viewpoints into meta model fragments: IT Consolidation, Business IT Alignment, Component Reuse, and Ownership. Note that for illustrative purposes these viewpoint specifications are simplified versions of the original viewpoints. The following elements of the viewpoint requirements template (VRT, cf. section 5) are omitted: representation of the object as-is, representation of the object to-be, modelers and information sources, model users and information targets. The information described in these elements is summarized in the example viewpoint relationship diagram shown in figure 6. The element “compatible approaches” of the VRT is not relevant for the viewpoints at hand. The “situation” to be addressed by the viewpoints is the architecture management and planning process of the partner company.

In all meta models shown in table 1 cardinalities and identifiers of relationships as well as attributes of classes are omitted. Figure 5 shows the integrated meta model from the four viewpoints (bold model elements) and further model elements originating from other viewpoints. All four meta model fragments are contained in the integrated meta model. The association between “application” and “system software” illustrates the need to modify meta model fragments during the integration process: Another viewpoint (Application Environment Management) raises further modeling requirements on the relationship between applications and system software that are not relevant in the context of the viewpoints “IT Consolidation” and “Component Reuse” presented here.

Figure 7 and Figure 8 illustrate how the meta model has been instantiated in models. Figure 7 shows processes and how these processes are mapped onto organizational units in a process landscape. Figure 8 shows the business IT alignment model relating processes and applications in a two dimensional matrix.

Figure 7: Example Model (Process Landscape)

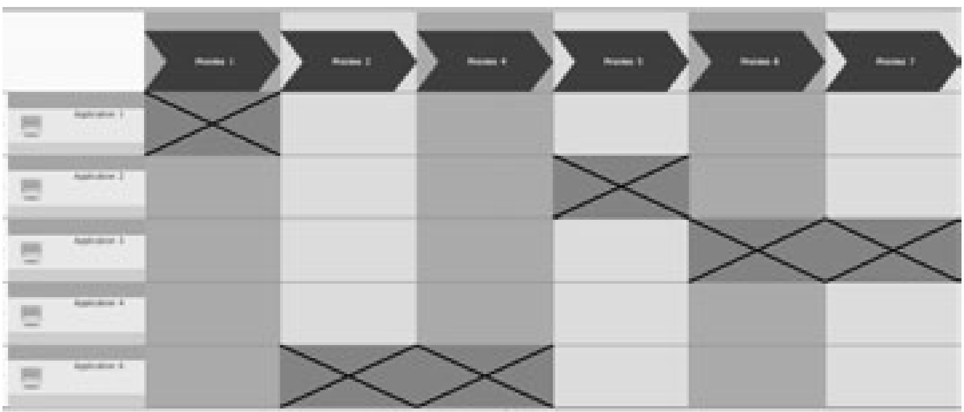


Figure 8: Example Model (Business IT Alignment Model relating Processes and Applications)