A Method for Aligning Business Process Modeling and Software Requirements Engineering

Daniel Weiß, Joerg Leukel, Stefan Kirn

Information Systems II
University of Hohenheim
Schwerzstr. 35
70599 Stuttgart
daniel.weiss@uni-hohenheim.de
joerg.leukel@uni-hohenheim.de
stefan.kirn@uni-hohenheim.de

Abstract: Software engineering provides a rich set of methods and tools that cover the entire engineering process. However, there still exist major problems in integrating methods that address single tasks and aspects only. This observation is in particular true for the interdependencies between business process modeling and software requirements engineering; respective methods originate from different though complementing areas, i.e., information systems and computer science. In this paper, we investigate the software engineering process from a method engineering perspective. The contribution is that we (1) propose a method for aligning business process modeling and software requirements engineering (ProQAM) and (2) show how this method fits in the overarching SIKOSA framework.

1 Introduction

In recent years, both industry and academia have proposed and investigated a foreseen paradigm shift in developing enterprise information systems towards industrialization. This shift aims at standardizing software engineering processes, reducing vertical integration, increasing reusability, and ultimately enhancing software quality [HO07]. A critical prerequisite for materializing this paradigm is the availability of highly integrated methods and tools that cover the entire *software engineering process*.

There exist, though, still *major problems in integrating methods* and their resulting models, since many methods have been designed for specific tasks within in the engineering process only. This observation is particular valid for business process modeling and software requirements engineering: Both areas concern requirements on enterprise information systems and focus similar though not equal aspects. For instance, business process modeling is essentially interested in control flows whereas requirements engineering takes a much broader view and emphasizes functional requirements. The seamless integration of methods from these two areas is also made complicated due to different origins, thus information systems vs. computer science, and therefore potentially inhomogeneous axioms, underlying theories, and terminology.

This paper analyzes the intersection of business process modeling and software requirements engineering from a *method engineering perspective* [BWH05]. Method engineering allows for designing methods by means of meta modeling [KK02; St96] describing the process aspect of the method and the language (or technique) used. Meta modeling contributes to systematically studying and decomposing the constitutive elements of existing or planned methods. Therefore, the structure of our research is that we first define the meta model language to be used, then specify meta models for all relevant domains and finally integrate the meta models into one method. The contribution to research is that we (1) propose the ProQAM-method (Process-oriented Questionnaires for Analyzing and Modeling Scenarios) for aligning business process modeling and software requirements engineering and (2) demonstrate how this method fits in the overarching framework of SIKOSA [WKK07].

The remainder of the paper is structured as follows. In section 2, we review related work. In section 3, we present the ProQAM method. In section 4, we describe the relationships of ProQAM to the overarching SIKOSA framework with respect to its usability in the succeeding phases of the software engineering process. Section 5 draws conclusions and outlines future work.

2 Related Work

The development of enterprise information systems calls for an integrated business process and software requirements analysis. Usually, this analysis takes place in a top-down manner which starts with business goals and decomposes them into software requirements of increasing detail. However, few approaches provide actually concrete means how to derive such requirements from business process models, e.g., [KL06a; OWS04; Sc01; BE01; EP00 or FS95].

The reason is that bridging both areas, business process modeling and software requirements engineering, has to overcome severe barriers such as (1) divergent syntax and semantics of respective language constructs and (2) different conceptualizations of business process quality respectively software quality. Therefore, transforming business process models into software requirements models often has to cope with the question how to maintain business semantics. Dedicated approaches for integrating business process and enterprise software requirements analysis can be found in [No04; KL06b]. Other studies, however, indicate that major shortcoming with regard to method integration still exist [BE01; KL06a].

Basically, two types of requirements can be distinguished: functional and non-functional requirements. Unfortunately, most current approaches and methods focus functional requirements only. Despite their increasing relevance for enterprise information systems and end-users, non-functional requirements have not attracted the same attention by researchers and therefore lack sound methodologies and software tools.

3 Proposed Method

In this section, we present the ProQAM method aiming at aligning business process modeling and software requirements engineering. By taking a method engineering perspective, we employ meta modeling for specifying the method. In general, a method consists of the following elements: (1) process model and (2) modeling language (technique) [Ba00]. Depending on the degree of formal semantics, such elements can be described either informally in documents or formally by means of respective meta models. Here, we chose the latter approach for all modeling languages; hence we design these languages by referring to existing languages and their respective meta models.

3.1 Meta Model Language

Requirements for a suitable meta model language are in particular the semantic power and determination [St96]. The semantic power describes how precise and differentiated the language can be used. The determination requires the absence of design freedom. In particular, such situations should be avoided in which various semantically equivalent modeling alternatives can be chosen for one concept [St96].

We apply the UML diagram type class diagram as model language for all ProQAM meta models [OMG07]. This diagram type meets the requirement of semantic power, since it provides a large number modeling concepts such as classes, attributes, generalization/specialization, aggregation, association, and association classes (see also [KK02]).

3.2 Process Model

The construction of semantically extensive and formal business process models requires sound knowledge regarding the actual business model. That means that detailed information concerning the involved actors, their primary activities, products, etc. is needed and calls for a systematic procedure. For this purpose, ProQAM proposes a process model, which comprises six phases and includes feedback loops:

1. Phase 1 (Value Chain Analysis) deals with *value chain analysis* and specification, and therefore with involved actors (in the sense of companies), their primary activities and produced goods and services.

2. In Phase 2 the information is used to build a *complete reference business process* model. In doing so, additionally elements, such as already available business processes reference models (e.g. [Sc98]) are considered.

3. Based on these results, Phase 3 (ProQAM Questionnaire Desgin) concentrates on designing the questionnaire to determine the *actual reference business process* model. The questionnaire involves structured and closed questions as they can be found in the method construction kit of the quantitative empirical social research [SHE05].

4. In Phase 4 (Business Process Elicitation), the actual business processes will be *elicitated* using methods such as observation, document analysis, or interviews.

5. Finally, the actual-business process will be *modeled*, *analyzed* (Business Process Analysis) and *compared* to the basic reference business process model.

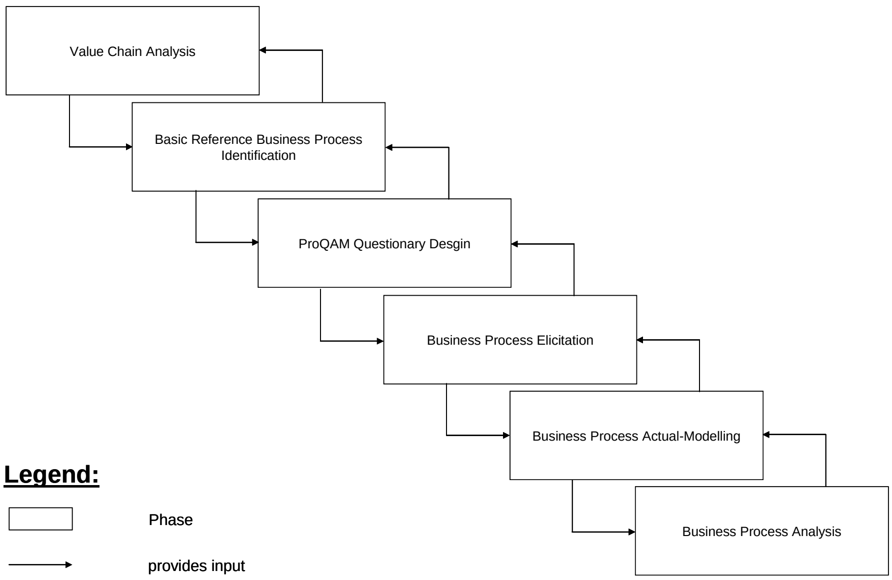

Figure 1: The ProQAM-Process Model

3.3 Modeling Languages

ProQAM comprises three modeling languages for organizations, business processes itself, and business process quality.

3.3.1 Organization

A poor understanding of the organizational setting is one main reason of software engineering project failures [CKI88]; thus a deep understanding of the needs, interests, priorities and abilities of the various actors is indispensable [Yu97]. Basic elements of an organizational model are therefore organizational units, functions, and roles [RM97; HFU94; BFO06]:

Organizational unit: The organizational structure is being formed by arranging multiple organizational units into a hierarchy or network. Such units can be differentiated in permanent and temporary ones (i.e., for projects).

Function: Organizational units are responsible for fulfilling one or many functions. These functions are also part of any process model. There exist various interdependencies between functions such as hierarchy, flow of information or goods, and equal properties which allow for grouping functions.

Role: By abstracting from concrete organizational units, we define a role as a set of minimal qualifications required for fulfilling a function (e.g., special skills); hence roles serve as an intermediary between units and functions.

Figure 1 depicts the meta model of organizational modeling in ProQAM.

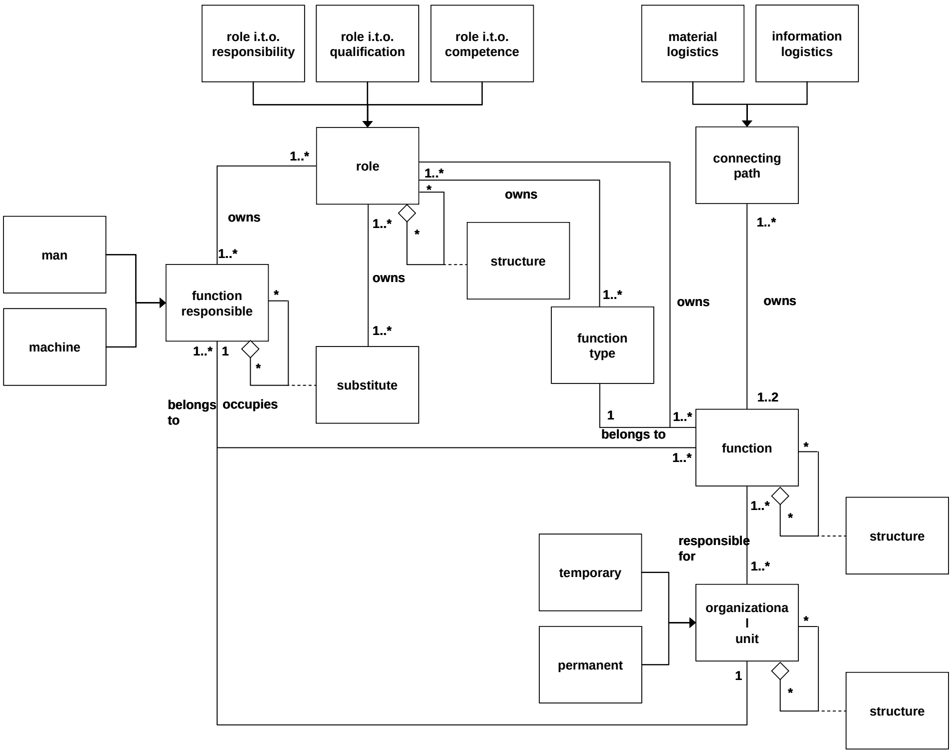

Figure 1: Organization Meta Model

3.3.2 Business Process

For describing a company’s business process landscape the proposed meta model (Figure 2) in will be applied. In the style of [Sc01], to its basic elements belong functions and events. Furthermore, the meta model considers the processed input and produced output, such as goods and services, as well as goals and function responsibles. While events describe defined states, functions represent the elementary functions to be processed. One or more event triggers or ends a function.

The process input, goods, services and information services, are essential for the business process and will be transformed during business process execution. Goals control the function executed by the function responsible.

Figure 2 provides the meta model of business process modeling in ProQAM.

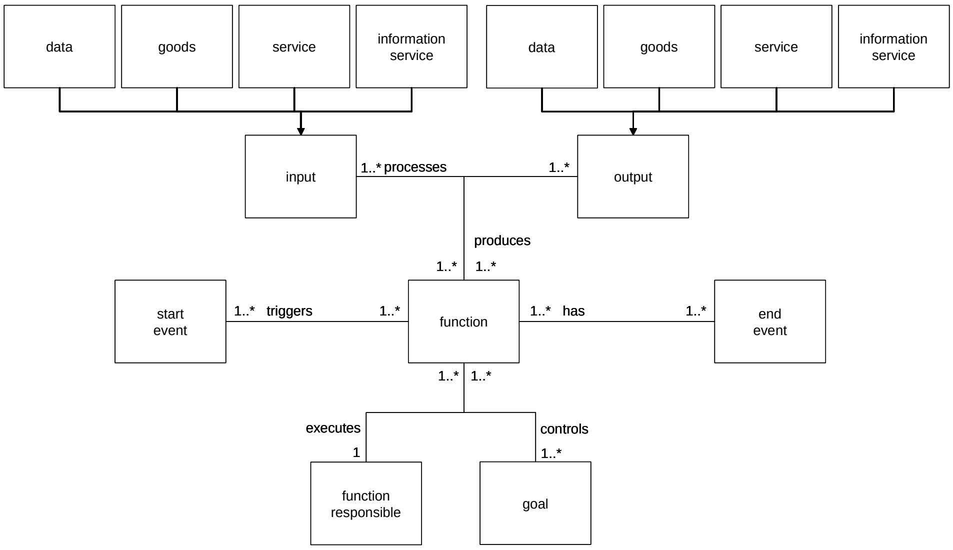

Figure 2: Business Process Meta Model

3.3.3 Quality

Similarly to the presented meta models, the quality meta model is also based on functions and goals as well. For specifying non-functional business process requirements, three types of quality attributes will be used. The relevant quality attribute at a time is determined by the business goals. A non-fulfilled non-functional requirement leads to quality defect and causes a business risk. The three types of quality attributes – structural, formal and content-addressed – define different dimension of business process quality. While structural quality attributes, such as compliance or complexity, affect to business process only indirectly, formal and content-addressed quality attributes influence the business process directly. These attributes address amongst others efficency, flexibility or integrity.

The meta model of quality modeling is shown in Figure 3.

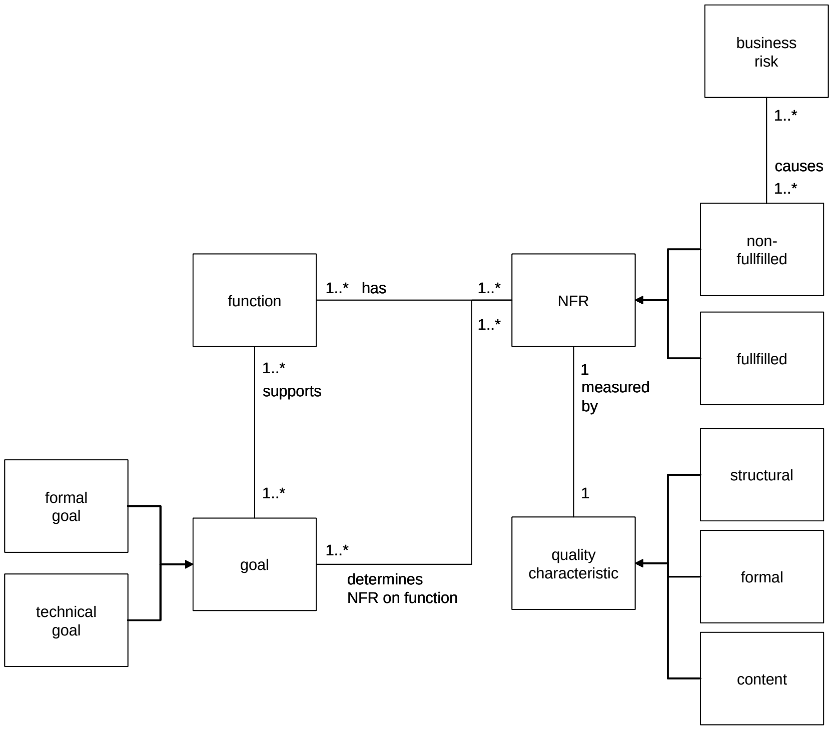

Figure 3: Quality Meta Model

4 Role of ProQAM in SIKOSA

4.1 The SIKOSA Framework

The SIKOSA framework explicitly addresses the intersection between software engineers and software users. Thus the phases of business process and software requirements engineering (in particular requirements analysis and specification), as well as software test, are focused. The latter is also addressed, as software test specifications ultimately result from software requirements. The SIKOSA framework is arranged around two pillars, (1) business process quality and (2) security monitoring, along which method ruptures are being reduced. SIKOSA provides the following methods besides ProQAM:

1. TORE (Task Oriented Requirements Engineering): The method supports the analysis and specification of functional software requirements regarding the user interface [PK03].

2. MOQARE (Misuse Oriented Quality Requirements Engineering): The method supports the combined analysis, specification and prioritization of functional and non-functional software requirements [HP07; HP05].

3. PAT[3] (Process, Automation, Testability, Transformation, Traceability): The method supports the specification of software tests on the basis of software requirements [WKK07].

4. RQP (Reverse Query Processing): The technology supports the production of relevant and reasonable business test data [BKL07; BKL06].

5. *VEP (*Vulnerability Evaluation Point): The tool supports monitoring software security during business process execution [WKK07].

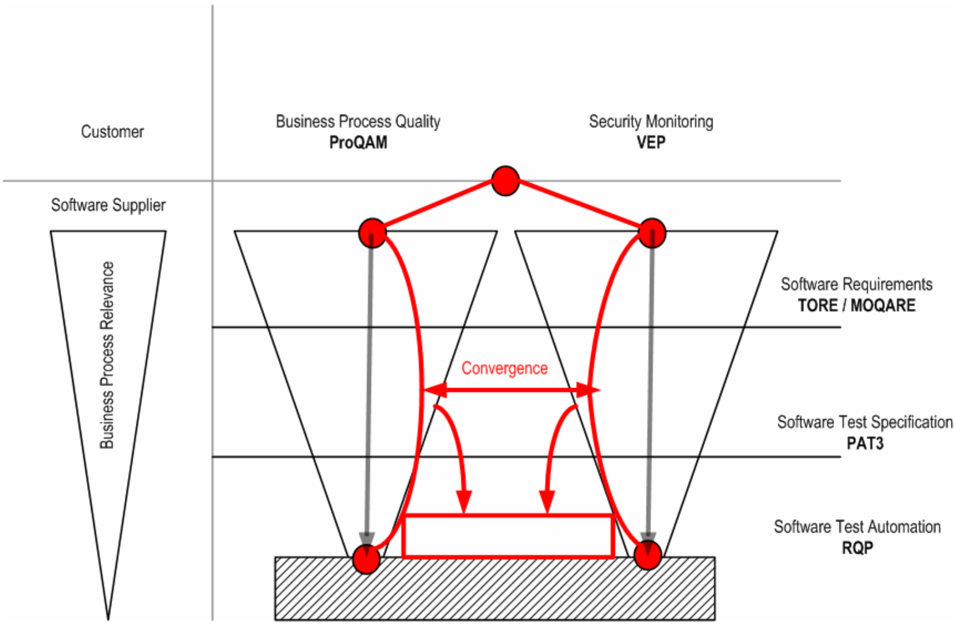

Figure 4: SIKOSA framework

The SIKOSA meta model, its process model and details regarding the single methods can be found in [WKK07; HPK06].

4.2 The ProQAM-SIKOSA interaction

Within the SIKOSA framework, ProQAM is interrelated with two other methods – MOQARE and TORE – and provides the input for the Requirements Engineering Phase. Such interrelation exists due to a common elements in the respective method meta models.

ProQAM-MOQARE:

- ProQAM provides for MOQARE not only the *formal* and *technical goals*, but also the *business risk*, which threatens the goals. The risk quantifies a possible *business damage* caused by an *IT quality defect*, and is defined as event risk * amount of damages.

- Furthermore, ProQAM provides the input for MOQARE concept *value*. A value represents the part of the systems to be protected and comprises not only IT-systems, but also all elements of the business process and organization meta model. Each value must be *protected* with regard to a specific *IT-quality attribute*. The choice of the IT-quality attribute depends on the *business quality characteristic*.

- The analysis and specification of the *function responsibles* and their *roles*, allows MOQARE systematically to identify possible *misusers* and *activities*. Such can be hackers, end-users, administrators or IT-systems.

- Finally, *non-functional business requirements* influence possible *counter-measures* in MOQARE, as counter-measures themselves can be new functional or non-functional requirements again.

ProQAM TORE

Since use case modeling in TORE happens from an end-user perspective, ProQAM delivers with function responsibles and their roles relevant information. The use cases describes in which way the users interact with the IT-system (activities, pre-conditions, post-conditions). Therefore, it is cruel to know which persons, and roles, interact with the system.

5 Conclusions

In this paper, we have presented a method for aligning business process modeling and software requirements engineering (ProQAM) and showed how this method fits in the overarching framework of SIKOSA. This method addresses an important shortcoming in current approaches for enterprise information systems engineering and respective software engineering approaches. Thus it contributes to better aligning basic concepts, terminology and actual modeling languages in the relevant domains.

By taking a method engineering perspective, we reused existing core elements of languages for business process modeling. The three ProQAM meta models are components of the overarching SIKOSA framework which provides a seamless integrated set of methods ranging from top-level business process modeling to, for instance, conducting software tests automatically based on models designed in the early development phases.

The ProQAM method has been evaluated in an industrial scenario within the project frame, whereas its general validity has not been studied so far. Another limitation is that – due to the plethora of existing methods, languages and tools for both business process modeling and software requirements engineering – we have had to limit the scope of the ProQAM modeling languages to a limited set of core elements.

Future work will concern evaluating ProQAM in other application scenarios, formalizing its questionnaire-oriented process model in terms of meta modeling as well as studying the relationship of business goal modeling and software specification techniques.

References

[Ba00] Balzert, H.: Lehrbuch der Software-Technik, Band 1, Software-Entwicklung. Heidelberg, 1996.

[BE01] Brücher, H.; Endl, R.: Erweiterung von UML zur geschäftsregelorientierten Prozessmodellierung. Heidelberg, 2002.

[BFO06] Bieser, T.; Fürstenau, H.; Otto, S.; Weiß, D.: Requirements Engineering. In (Kirn, S.; Herzog, O.; Lockemann, P.; Spaniol, O. Hrsg.): Multiagent Engineering. Theory and Applications in Enterprises. Heidelberg, 2006, S. 359-381.

[BKL06] Binnig, C.; Kossmann, D.; Lo, R.: Testing database applications. In: Proceedings of the 25th ACM SIGMOD International Conference on Management of Data / Principles of Database Systems. Chicago, 2006, S. 739-741.

[BKL07] Binnig, C.; Kossmann, D; Lo, R.: Reverse Query Processing. Generating Meaningful Test Databases. In: ICDE 2007. Istanbul, 2007.

[BWH05] Braun, C.; Wortmann, F.; Hafner, M.; Winter, R.: Method Construction - A Core Approach to Organizational Engineering. In: ACM Symposium on Applied Computing. Santa Fe, 2005, S. 1295-1299.

[CKI88] Curtis, B.; Krasner, H.; Iscoe, N.: A Field Study of the Software Desgin Process for Large Systems. In: Communications of the ACM 31 (1988) 11, S. 1268-1287.

[EP00] Eriksson, H.-E., Penker, M.: Business Modelling with UML. New York, 2000.

[FS95] Ferstl, O.; Sinz, E.: Der Ansatz des semantischen Objektmodells zur Modellierung von Geschäftsprozessen. In: WIRTSCHAFTSINFORMATIK 37 (1995) 3, S. 209-220.

[HFU94] Hill, W.; Fehlbaum, R.; Ulrich, P.: Organisationslehre. Stuttgart, 1994.

[HO07] Heinzl, A.; Oberweis, A.: Industrialisierung der Softwarenetwicklung – Eine Bestandsaufnahme. In: WIRTSCHAFTSINFORMATIK 49 (2007) 3, S. 157-158.

[HP07] Herrmann, A.; Paech, B.: MOQARE: Misuse-oriented Quality Requirements Engineering. In: Requirements Engineering Journal, to be published.

[HP05] Herrmann, A.; Paech, B.: Quality Misuse. In: Proc. 11th Int. Workshop on Requirements Engineering: Foundation of Software Quality – REFSQ 05, Essener Informatik Beiträge, Band 10, 2005; S. 193-199.

[HPK06] Herrmann, A.; Paech, B.; Kirn, S.; Kossmann, D.; Müller, G.; Binnig, C.; Gilliot, M.; Illes, T.; Lowis, L.; Weiß, D.: Durchgängige Qualität von Unternehmenssoftware. In: Industrie Management, 6, 2006; S. 59-61.

[KK02] Karagiannis, D.; Kühn, H.: Metamodelling Platforms. In LNCS 2455, Berlin, 2002, S. 451-464.

[KL06a] Korherr, B.; List, B.: Aligning Business Processes and Software Connecting the UML 2 Profile for Event Driven Process Chains with Use Cases and Components. In: Proc. 18th Int. Conf. on Advanced Information Systems Engineering CAiSE´06. Luxembourg, 2006, S. 19-22.

[KL06b] Korherr, B.; List, B.: A UML 2 Profile for Event Driven Process Chains. A UML 2 Profile for Business Process Modelling. In: Proc. 1st Int. Workshop on Best Practices of UML at the 24th Int. Conf. on Conceptual Modeling ER 2005. Klagenfurt 2005.

[No04] Noran, O.S.: Business Modelling: UML vs. IDEF. http://www.cit.gu.edu.au/~noran/Docs/UMLvsIDEF.pdf, Abruf am 2006-10-12.

[OMG07] Object Management Group (OMG): Unified Modeling Language: Infrastructure (version 2.1.1 / 07-02-06). http://www.omg.org/docs/formal/07-02-06.pdf, Abruf am 2007-03-16.

[OWS04] Oestereich, B.; Weiss, C.; Schröder, C.; Weilkiens, T.; Lenhard, A.: Objektorientierte Geschäftsprozessmodellierung mit der UML. Heidelberg, 2004.

[PK03] Paech, B.; Kohler, K.: Task-driven Requirements in object-oriented Development. In (Leite, J.; Doorn, J., eds.): Perspectives on Requirements Engineering, 2003.

[RM97] Rosemann, M.; zur Muehlen, M.: Modellierung der Aufbauorganisation in Workflow-Management-Systemen. Kritische Bestandsaufnahme und Gestaltungsvorschläge. In (Jablonski, S. Hrsg.): Proceedings of the EMISA-fachgruppentreffen. Darmstadt 1997, S. 100-118.

[Sc01] Scheer, A.-W.: ARIS – Modellierungsmethoden, Metamodelle, Anwendungen. 4. Aufl., Berlin, 2001.

[Sc98] Scheer, A.-W.: Referenzmodelle für industrielle Geschäftsprozesse. 2 Aufl., Berlin, 1998.

[SHE05] Schnell, R.; Hill, P.; Esser, E.: Methoden der empirischen Sozialforschung. 7. Aufl., München, 2005.

[St96] Strahringer, S.: Metamodellierung als Instrument des Methodenvergleichs: Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Aachen, 1996.

[WKK07] Weiß, D.; Kaack, J.; Kirn, S.; Gilliot, M.: Die SIKOSA-Methodik. In: WIRTSCHAFTSINFORMATIK 49 (2007) 3, S. 188-198.

[Yu97] Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering. Washington, 1997, S. 226-235.