Method Versus Model – Two Sides of the Same Coin?

Robert Winter, Anke Gericke, and Tobias Bucher

Institute of Information Management, University of St. Gallen,
Müller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
{Robert.Winter,Anke.Gericke,Tobias.Bucher}@unisg.ch

Abstract. This article analyzes the state-of-the-art regarding the development of generic methods and reference models. The analysis shows that the related research disciplines, method engineering and reference modeling, tend to converge. Furthermore, it shows that the differentiation between generic methods and reference models should not be maintained because both artifact types feature activity-oriented elements as well as result-oriented elements. Depending on the artifact type, however, generic methods and reference models vary regarding the relative importance of the activity view and the result view. A generic problem solution (generic term for methods and reference models) can be interpreted as a sequence of activities which aim at the development of results. The insights into the commonalities among generic problem solutions provide the opportunity to define a unified design process in the field of design science research. Implications and unification challenges that are related to such a unified design process are presented at the end of the paper.

Keywords: Design Process, Method Engineering, Reference Modeling.

1 Introduction

Information systems (IS) researchers follow two main research approaches: behavioral research and design science research (DSR) [20, p. 76]. In contrast to behavioral research which is primarily aimed at advancing the body of knowledge through theory building, DSR is a problem solving paradigm which "has its roots in engineering" [20, p. 76]. The ultimate goal of the DSR approach is the development of useful artifacts that bear the potential to solve relevant IS problems [30, p. 253]. In this article, IS are generally understood as socio-technical systems. Socio-technical IS comprise all persons, business processes, software and information technology infrastructure that process data and information within an organization [cf. e.g. 2; 8; 38; 43].

March and Smith [30, p. 256 ff.] have established a widely accepted taxonomy of artifact types of DSR: constructs, models, methods and instantiations. In addition, design theories (relating to the design of artifacts, as opposed to general theories from the behavioral research paradigm) have been discussed as an extension of the DSR artifact taxonomy lately [cf. e.g. 25; 42].

Many European DSR communities are focusing on two specific artifact types: On the one hand, method engineering is addressing the development of generic methods and their adaptation in order to solve relevant IS problems. On the other hand, reference

A. Albani, J. Barjis, and J.L.G. Dietz (Eds.): CIAO!/EOMAS 2009, LNBIP 34, pp. 1–15, 2009.

modeling is aimed at the development of reusable conceptual models and their adaptation to solve relevant IS problems. However, as an analysis of contributions to the 2006 and 2007 International Conferences on Design Science Research in Information Systems and Technology (DESRIST) shows, research has mainly focused on the development of instantiations [10, p. 42]. Thus, we want to bridge this gap by analyzing the construction of generic methods and reusable conceptual models within their respective disciplines. Results of our analysis will represent the actual state-of-the-art regarding the development of both artifact types. Using this as a basis, topics for further research within both disciplines will be proposed. Addressing these research issues can contribute to the ongoing development of the method engineering and reference modeling discipline.

The remainder of the article at hand is structured as follows: In the following section, we present our analysis of related work on method engineering and reference modeling. The analysis shows that these two research disciplines are converging, in particular regarding "design knowledge". A convergence can also be observed in "artifact construction". These observations are outlined in the third section. They lead to the conclusion that generic methods and reusable conceptual models are two views on the same underlying object. This hypothesis is then taken up in the fourth section in which the relationship between generic methods and reusable conceptual models is analyzed in-depth. Within that section, a taxonomy of methods and models is presented. Based on that foundation, the fundamentals for a unified construction process for generic methods and reusable conceptual models are proposed. Since our hypothesis still holds true, we discuss some consequences for such a unified design process in the fifth section. The article closes with a summary and an outlook.

2 State-of-the-Art Analysis

Method engineering is concerned with the development of generic methods; reference modeling addresses the construction of reusable conceptual models. A review of the state-of-the-art in both disciplines is presented in the following. For this review the focus will be laid on the problem definition, construction/development and evaluation phases of the DSR process [cf. 32, p. 91 ff.].

2.1 Method Engineering

The method engineering (ME) discipline is concerned with the processes of constructing, adapting and implementing methods for the design of information systems [7, p. 276]. According to Brinkkemper, a method is "[…] 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 products" [7, p. 275 f.]. Such methods can be denoted as generic methods, as they are not restricted to solve only one specific problem, but rather address a class of (similar) design problems. In addition to this understanding of the term "generic method", different method definitions exist that particularly differ in respect to the method meta model [cf. 6]. All authors agree that a generic method consists of several activities and corresponding results [6, p. 1297]. Although activities and results are closely related to each other, they are often represented by two different models: activities and their sequence are represented by a procedure model while a result/deliverable model is used to represent results. Recently, so called process-deliverable diagrams [40, p. 36] have been proposed to jointly represent activities/activity sequences as well as results/deliverables and their relationships. A process-deliverable diagram is a combination of an UML activity diagram and an UML class diagram. In addition to activities and results (and the respective relationships), roles or techniques are often regarded as method meta model elements, too [6, p. 1297].

In order to be applicable for IS development, generic methods need to be adapted to the specific characteristics of the problem situation. This issue has been addressed in the ME discipline by proposing different construction processes for the development of so called situational methods [cf. e.g. 7; 21; 34; 39]. In order to provide a conceptual structure for these approaches, Bucher et al. [9, p. 35] and Bucher and Winter [10, p. 47 f.] suggest to differentiate situational method configuration and situational method composition. The distinguishing mark of situational method configuration is the adaptation of a so called base method against the background of a specific problem situation [9, p. 35]. By contrast, the fundamental idea of situational method composition is the selection and orchestration of method fragments with respect to the specifics of a problem situation [9, p. 35 f.]. Unlike situational method configuration, the composition process is not aimed at configuring one single base method, but at combining and aggregating several method fragments in order to establish new constructional results. Situational method composition is widely used and discussed in detail in the scientific literature [cf. e.g. 5, p. 6 f.].

Regarding these two different construction/development processes, the question arises how the problem situations can be characterized in which the methods will be used. Although the necessity for such a characterization of the problem situation (as part of the problem definition phase) has often been stated, there are only few approaches for defining a problem situation [9, p. 36]. Again, two different types can be differentiated. On the one hand, there are approaches that present different, predefined contingency factors such as “size of the project”, “number of stakeholders” or “technology used” (cf. e.g. [24, p. 68 ff.] and [41], cited after [35, p. 12]). On the other hand, Bucher et al. [9] and Mirbel und Ralyté [31] characterize a situation e.g. by means of so called context type factors and project type factors [9, p. 37 ff.]. In contrast to the first approach, these factors are not predefined, but instead have to be identified individually for each and every problem situation and/or application domain.

Both the development of generic methods and the mandatory description of the problem situation have already experienced a wider research interest. In contrast, only few researchers have addressed the evaluation of methods up to now. Being the only contribution to this field to our knowledge, Pfeiffer und Niehaves [33, p. 5] present different evaluation approaches for the evaluation of methods such as case studies, action research or surveys.

2.2 Reference Modeling

Reference modeling is an IS research discipline dealing with the construction and application of reusable conceptual models, so called “reference models” [45, p. 48 ff.]. A reference model contains recommendations or references which can be used for the design of IS or the construction of other models [12, p. 35; 45, p. 48 f.; 46, p. 586 f.]. In addition to the reference character and to reusability (which are related to each other, cf. [44, p. 31 ff.]), there are other characteristics of reference models that are discussed in the literature as well. One of these characteristics is universality. Universality means that reference models should be valid solutions for an (abstract) class of problems [12, p. 35; 46, p. 584]. Over the past years, several procedure models have been developed that support the construction of reference models [cf. e.g. 15, p. 22 f.; 46, p. 591 ff.]. They do not differ significantly from each other and comprise the three generic construction phases outlined in section 2.

Up to now, only few contributions address the description of the problem situation in reference modeling [cf. e.g. 5, p. 7]. In contrast, numerous articles address the development phase of the construction process. For the reason of being adaptable to different problem situations when applying the reference model, the reference model has to be equipped with adaptation mechanisms during the development phase [45, p. 49]. Moreover, recommendations on how to adapt or how to use the reference model have to be provided [22]. Regarding the adaptation mechanisms, so called generating and non-generating approaches can be differentiated [17, p. 1]. Generating adaptation mechanisms are also referred to as configuration mechanisms and can be divided into (1) model type selection, (2) element type selection, (3) element selection, (4) synonym management and (5) presentation variation [3, p. 221 f.; 23, p. 136 ff.]. With respect to non-generating adaption mechanisms, aggregation, specialization, instantiation and analogy can be differentiated [cf. e.g. 44, p. 284 ff.; 45, p. 58 ff.]. After developing a reference model, an evaluation should be conducted in order to prove the utility of the model [13, p. 81]. In principle, such an evaluation can refer to the construction process itself or to the product of this process (i.e. the reference model). For both types of evaluation, different evaluation methods are available, such as the guidelines of modeling [37], ontological evaluation [14, 19] or evaluation based on case studies [13, p. 83]. The evaluation framework proposed by Fettke and Loos [13] provides an overview and systematization of different evaluation methods.

3 Convergence of Method Engineering and Reference Modeling

Following Hevner et al. [20, p. 87], different types of contributions can be differentiated in DSR: On the one hand, there are contributions in the area of “design knowledge” (“design construction knowledge” and “design evaluation knowledge”). On the other hand, “design artifacts” are considered as valid DSR contributions, too.

While analyzing the state-of-the-art of both disciplines, a convergence of both disciplines in respect of design knowledge can be observed. This is especially true for the area of design construction knowledge. Using this as a basis, we analyze whether such a convergence can be observed in the area of design artifacts as well. Thereafter, the conclusions drawn from these findings are presented, resulting in a proposed hypothesis that will be scrutinized in section 4.

3.1 Convergence in Respect of Design Knowledge

Researchers from the ME discipline [cf. 36; 47] as well as from the reference modeling discipline [cf. 4; 5] ask for the transfer of developed concepts to the respective "counterpart". Based there-on, several efforts have been undertaken to transfer existing research results in the different phases of the construction process, i.e. problem definition, development and evaluation. These efforts are presented in the following.

In reference modeling, only few approaches exist that deal with the specification of the problem situations in which a reference model should be used [5, p. 7]. By contrast, this topic has been addressed more intensively in ME. Although Schelp and Winter [36; 47] ask for the transfer of these results to the reference modeling discipline, contributions are still missing that describe how the specification of problem situations can be transferred to reference modeling in detail.

With respect to the transfer of adaptation mechanisms from reference modeling to ME (development phase), some first research results were achieved. Based on the assumption that both generic methods and reference models can be represented with the help of modeling languages, generating adaptation mechanisms (i.e. the configuration approach) have been formally transferred to generic methods [5, pp. 1, 7]. The applicability of element selection (one specific type of configuration) in ME could be shown on a procedure model as well [36, p. 569]. In this context, researchers still have to examine whether the other types of configuration (e.g. model type selection or element type selection) can also be applied to generic methods. On the contrary, the non-generating adaptation mechanisms instantiation and analogy have been considered scarcely in the ME discipline [4, p. 88; 5, p. 10].

The literature analysis (see section 2) shows that, although an evaluation is asked for in both research disciplines, this issue has hardly been addressed in research yet. That is why no contributions can be identified which deal with the transfer of evaluation approaches from one discipline to the other.

The analysis in respect of the convergence of both research disciplines in the area of design knowledge shows that the transfer of different approaches from ME to reference modeling and vice versa has already been done successfully or is at least intended. Using this as a basis, we analyze in the following whether or not this convergence can be observed for design artifacts, too.

3.2 Convergence in Respect of Design Artifacts

In order to determine whether the proposed convergence of ME and reference modeling can also be recognized regarding constructed artifacts, we will analyze case examples from current publications. For the identification of such case examples, we focus on an article of Bucher and Winter [10] that classifies contributions to the 2006 and 2007 International Conferences on Design Science Research in Information Systems and Technology (DESRIST) with respect to the type of artifact developed/presented. Based on this article, we select all articles that are classified as either method or model (see Table 1). However, we do disregard articles that are assigned to more than one type of artifact. We choose the article of Bucher and Winter [10] as well as the underlying DESRIST conference proceedings because this research community possesses a high research culture homogeneity – as they follow the DSR paradigm. Besides that, these proceedings enable us to take recent publications into account. Table 1 gives an overview about the chosen case examples.

Table 1. Case Examples

No.	Case Example	Predominant Type of Artifact
1	Arazy et al. 2006: Social Recommendations Systems: Leveraging the Power of Social Networks in Generating Recommendations [1]	Model
2	Gorla and Umanath 2006: On the Design of Optimal Compensation Structures for Outsourcing Software Development and Maintenance: An Agency Theory Perspective [16]	Model
3	Kunene and Weistroffer 2006: Design of a Method to Integrate Knowledge Discovery Techniques with Prior Domain Knowledge for Better Decision Support [26]	Method
4	Zhao 2006: Selective Encryption for MPEG-4 FGS Videos [48]	Method

Analyzing these case examples, it can be recognized that the development of (reference) models is predominant in the first two articles [see 1; 16]. In addition, both articles contain simple activity recommendations in the form of *Use the proposed model for the development of social filtering systems* [cf. 1, p. 320] or *Use the proposed model to optimize your compensation structures for outsourcing software development and maintenance* [cf. 16, p. 660]. These activity recommendations take the form of recommendations that are normally presented by generic methods. In contrast to articles one and two, generic methods are developed in articles three and four [see 26; 48]. Although activity recommendations are predominant, results (as normally presented with the help of reference models) are explicated as well, e.g. by giving examples for the results of some of the actions that are part of the method [cf. 26, p. 348 ff.; 48, p. 607 f.]. Summarizing this analysis, it can be stated that although one artifact type is always pre-dominant, aspects from both generic methods and reference models can be identified in all case examples simultaneously.

3.3 Intermediate Findings

In ME as well as in reference modeling, several topics such as artifact construction processes, contingency approaches for the adaptation of generic/reusable artifacts, mechanisms for adaptation, etc., have been developed separately (see section 2). Recently, both disciplines have increasingly cross-fertilized each other, resulting in the transfer of different concepts/topics from one discipline to the other (see above). This is not only true within the area of design knowledge of both disciplines. Rather, the convergence of ME and reference modeling can also be observed when looking at the actual construction of artifacts. Thus, our findings suggest that generic methods and reference models are somehow similar and/or related to each other. We therefore propose the following hypothesis:

> *Generic methods and reference models represent different views on the same underlying object.*

In the following, we justify this hypothesis using argumentative analysis. Our ultimate goal is to understand the relationship between generic methods and reference models. Insights into this relationship can serve as a foundation for a unified design process for both generic methods and reference models.

4 Discussion of the Hypothesis

In order to justify our hypothesis, we first introduce a taxonomy that compares generic methods and reference models. Based on that foundation, the relationship between generic methods and reference models is then explicated, allowing for the proposition of a unified design process.

4.1 Positioning Generic Methods and Reference Models in a Model Taxonomy

Based on the argumentation of Becker et al. [5, p. 1] that both generic methods (especially procedure models as constituent elements of methods) and reference models can be represented by means of models[1], we develop a taxonomy in which generic methods and reference models can be positioned.

IS models can be divided into description models, explanatory models and design models [27, p. 20]. Description models and explanatory models are understood as descriptive models whereas design models are considered to be prescriptive or instructional [28, p. 284]. The latter thus possess a recommendatory character. Due to the fact that generic methods as well as reference models express recommendations, these two model types are assigned to the category of design/prescriptive models. We will abstain from discussing descriptive models in the following.

In a second step, the prescriptive models can be further subdivided. A differentiation can be made regarding the way of recommendation: On the one hand, recommendations can refer to a (design) activity; on the other hand, they can refer to the result of that activity [28, p. 284]. Following this argumentation, generic methods/procedure models are assigned to the category of models that prescribe recommendations for activities. This is due to the fact that they provide instructions and recommendations about how to obtain a solution of an IS problem. On the contrary, reference models can be assigned to the category of models that represent a recommendation for a design result.[2]

In addition to generic methods and reference models, there are prescriptive models that are specific. This category of models includes e.g. project plans as specific activity recommendations or data and functional models of IT solutions as specific result recommendations. Those models have been exclusively developed for a single, specific

---

[1] In this context, the term “model” does not refer to a reusable conceptual model or the model term as defined by March and Smith [30]. Instead, it refers to the general meaning of a model as a representation or abstracted view of the underlying object.

[2] Reference models always express recommendations for design results. This is true for all reference models, irrespective of the reference model being e.g. a reference process model (action-oriented) or a reference model of an organizational structure (state-oriented). However, we will not focus on this content-related differentiation. Regardless of the reference model’s design object being action-oriented (e.g. a process) or state-oriented (e.g. an organizational structure) it always provides a result recommendation for that design object.

problem. The distinguishing mark between generic methods and reference models on the one hand and those specific models on the other hand is the “intention for re-use”. Generic methods and reference models possess a recommendatory character and are valid for an (abstract) class of design problems. Moreover, they are explicitly designed to be re-used. On the contrary, specific models express recommendations for the solution of one specific design problem only. They are not intended for re-use.

Fig. 1 summarizes the arguments and depicts the proposed taxonomy for prescriptive models.

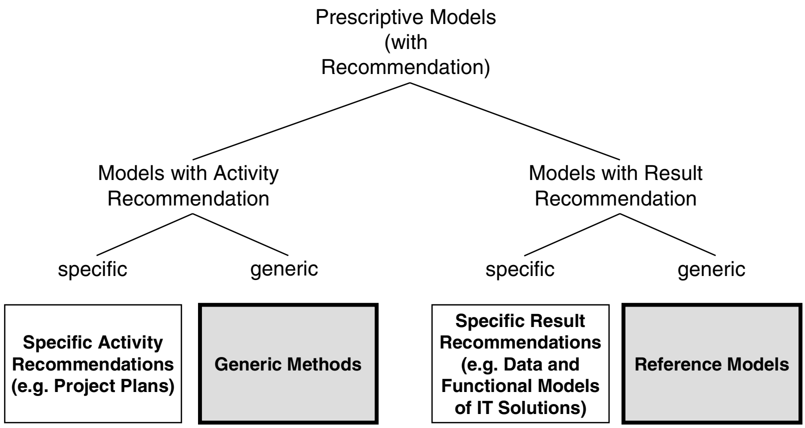


Fig. 1. Taxonomy of Prescriptive Models

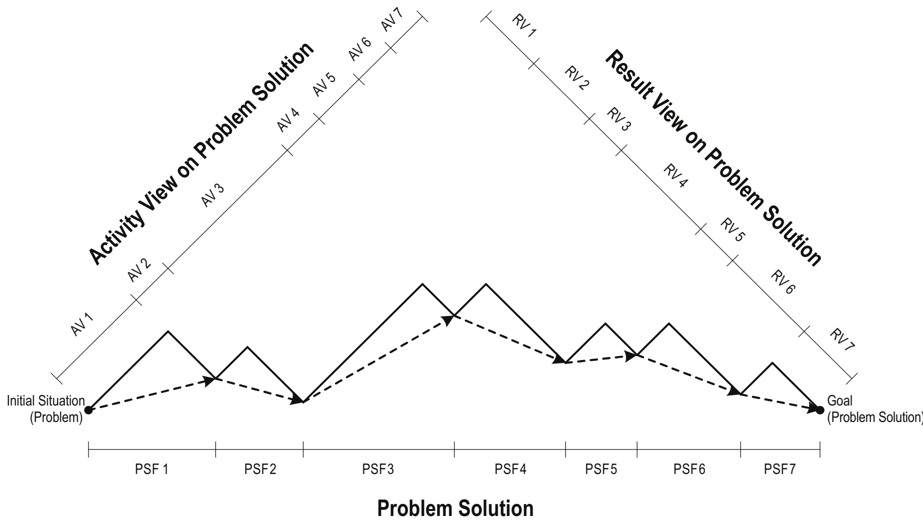
4.2 Towards a Unified Design Process for IS

According to the taxonomy presented previously, generic methods and reference models can be differentiated primarily regarding their type of recommendation (activity vs. result). However, the literature analysis (see section 2) as well as the analysis in section 3 implicate that such a stringent differentiation cannot be maintained.

As outlined before, generic methods also describe possible results of the recommended activities [cf. e.g. 26; 48]. Similarly, reference models provide activity recommendations, e.g. on how to adapt the model to and/or on how to use the model in a certain problem situation [cf. e.g. 1; 16]. This argumentation leads to the conclusion that an activity view and a result view can be defined for both generic methods and reference models. Depending on the type of artifact (generic method vs. reference model), however, they vary regarding the relative importance that these two views have. Thus, we will denote each artifact that possesses both an activity and a result view as a problem solving artifact in the following.

Based on this argumentation, a problem solving artifact (or, rather, a problem solution) can be interpreted as a sequence of (partial) activities which develop (partial) results in order to solve a certain class of problems. Hence, problem solutions represent “means-ends relations” [11].

Fig. 2 illustrates this understanding of the problem solution process.



AV = Problem Solution Fragment's Activity View
RV = Problem Solution Fragment's Result View
PSF = Problem Solution Fragment

Fig. 2. Problem Solution as a Sequence of Fragments that Have Both Activity and Result Character

Using this as a basis, we can differentiate two views on a problem solution:

- One problem solution view focuses on activities and can be designated as generic method. In detail, a generic method is understood as an artifact that represents a means-ends relation. The focus on activities inherent to this problem solution view is understood in the way that a generic method exactly describes how to create a solution for a problem/problem class, whereas the corresponding results are only implied or described rudimentarily.
- Another problem solution view focuses on results and can be designated as reference model. Equivalent to a generic method, a reference model is an artifact representing a means-ends relation. The focus on results inherent to this problem solution view is understood in the way that a reference model exactly characterizes a solution to a problem/problem class whereas the activities needed to solve the problem are only implied or described rudimentarily.

5 Consequences for a Unified Design Process

The arguments brought forward in the previous section support our hypothesis that generic methods and reference models are two sides of one and the same coin. This insight could and should lay the foundation for the definition of a unified design process for generic methods and reference models in DSR.

A similar development towards unified design can be observed in traditional engineering. In the 1960ies and 1970ies, different construction methods and processes have been developed in different research fields of the engineering discipline. Such "specific" construction methods and processes have been amalgamated into a "universal design theory" [cf. 18; 29] which is comprised of "findings and knowledge about design from different scientific and engineering disciplines in a consistent, coherent and compact form" [29, p. 203]. Domains integrated into the universal design theory are, for example, chemistry, chemical engineering, material science or technical biology.

Similar to the benefits outlined for the engineering discipline [29, p. 209], a unified design process for the construction of problem solutions in DSR would allow to achieve different benefits:

- The design process for the development of generic methods and reference models will become more efficient and reliable. This benefit can be realized by a design process that allows for the definition of construction processes, contingency approaches and mechanisms for artifact adaptation that are valid for both generic methods and reference models.
- If there is more than one possible tuple (activity, result) for a specific step in the problem solution process, the unified design process supports the evaluation and appreciation which of the possible tuples is more appropriate to contribute to the problem solution.
- Based on such a design process, different assumptions, procedures and outcomes of an artifact construction are made comparable and the design process becomes easier to control.
- With the definition of a unified design process, research efforts are reduced because several questions do not have to be discussed individually for each type of artifact.
- Based on such a unified design process, learning effects will be achieved that result from interdisciplinary knowledge acquisition. Furthermore, the construction of IS artifacts within interdisciplinary research teams is supported.

After explaining the advantages of a unified design process for generic methods and reference models, we will analyze the consequences with respect to the three construction phases: problem definition, development and evaluation (see section 2).

A problem solution, i.e. a generic method or a reference model, is used to solve relevant IS problems. As a precondition, it is necessary to describe the problem situation. In this context, two different assumptions about the consequences within a unified design process can be made: On the one hand, it can be reasonably assumed that existing problems are independent of the artifact type with which the problems are to be solved. Following this argumentation, research questions that address the description and specification of problems can be answered on a superordinate level for problem solutions. Hence, such research questions do not have to be considered individually in the ME discipline and in the reference modeling discipline. Instead, existing research results from both disciplines can be used to develop either a generic method or a reference model. This is true for research questions addressing the definition of problem situations in general (design knowledge), and problem descriptions for concrete application domains (design artifact). On the other hand, it can be assumed that different problem descriptions are necessary with respect to the kind of recommendations (activity/result) that will be developed. For the development of recommendations for activities, it might be necessary to describe the initial and the target state of a situation, whereas for the development of recommendations for results it is probably sufficient to describe only one state (either initial or target).

In the development phase of the construction process, research results in the area of design construction knowledge of generic methods and reference models can be used in both disciplines. Thus, research results regarding a procedure (sequence of activities) are not only valid for activity-oriented recommendations in the context of generic methods, but also for activity-oriented recommendations in the context of reference models. In turn, this is also true for result-oriented recommendations which are also valid in the context of both generic methods and reference models. Moreover, with the help of such a dedicated examination of the recommendatory character (activity vs. result), it will be considerably easier to use research results which have been gained in reference modeling with particular respect to results for the development of activities in ME instead. To make an example, the application of the adaptation mechanism "element type selection" of reference modeling to activities in ME could be introduced. Although such an application has already been conducted formally, the question arises under which terms and conditions this is possible for the content as well. For example, the utility of a method might be questionable if the element type "activity" was extracted from the method's procedure model. The clarification of such questions will be the basis for the construction of a unified design process. In addition to the consequences within the area of design knowledge, existing result-oriented recommendations could be integrated more frequently into the construction of activity recommendations and vice versa ("design artifact").

Finally, the consequences for a unified design process with respect to the evaluation of generic methods and reference models have to be analyzed. Since the research field of evaluation is not well-developed for both artifact types serious limitations have to be taken into account when making unification efforts. With respect to the design evaluation knowledge, it might be possible to use evaluation methods from ME and/or reference modeling to evaluate either a generic method or a reference model. Analogous to the development phase, we assume that evaluation results for activity recommendations are not only valid for generic methods, but also for reference models. This is also true for the evaluation of result recommendations for reference models that are also valid for generic methods ("design artifact").

6 Conclusion and Outlook

In the article at hand, we analyze the state-of-the-art of ME and reference modeling. These two disciplines of DSR for IS deal with the construction of generic methods and reference models, respectively. By analyzing the body of literature of both disciplines, a convergence of ME and reference modeling becomes evident. This is not only true in respect of the design knowledge of both disciplines, but also for the construction of concrete artifacts – as four case examples show. Thus, we propose the hypothesis that generic methods and reference models are two sides of one and the same coin. This hypothesis holds true as the argument can be brought forward that both generic methods and reference models can be viewed as a complex activity (procedure model, including all activities) as well as a complex result (including all intermediate/partial results). However, both artifact types vary with respect to their focus on the activity view and the result view, respectively. Following this argumentation, consequences for a unified design process have been presented. These arguments form the basis for further research activities.

In future research projects that deal with the development of generic methods and/or reference models, experience should be collected regarding the application of the design process of both ME and reference modeling. Those experiences will form the basis for the development of a unified design process that incorporates distinct parts that are valid for both artifact types as well as other parts that can only be applied under certain conditions.

Before developing such a unified design process, an in-depth analysis of the arguments brought forward in this article has to be performed, including formal analyses. In addition, the "artifact" term should be revisited as a consequence of our analysis. The strict definition of an artifact typology, as e.g. presented by March and Smith [30], might not be appropriate any more. Instead, a more general understanding of the term "artifact" should be developed. This can be achieved, for example, by defining a generic artifact as follows:

> *A generic artifact consists of language aspects (construct), aspects referring to result recommendations (model), and aspects referring to activity recommendations (method) as well as instantiations thereof (instantiation).*

In this respect, contributions should be called in that explicitly analyze the relations between these four aspects and put them in context to theories.

References

1. Arazy, O., Kumar, N., Shapira, B.: Social Recommendations Systems: Leveraging the Power of Social Networks in Generating Recommendations. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, pp. 310–328 (2006)
2. Bacon, C.J., Fitzgerald, B.: A Systemic Framework for the Field of Information Systems. ACM SIGMIS Database 32, 46–67 (2001)
3. Becker, J., et al.: Configurable Reference Process Models for Public Administrations. In: Anttiroiko, A.-V., Malkia, M. (eds.) Encyclopedia of Digital Government, pp. 220–223. Idea Group, Hershey (2006)
4. Becker, J., Janiesch, C., Pfeiffer, D.: Reuse Mechanisms in Situational Method Engineering. In: Ralyté, J., et al. (eds.) Situational Method Engineering – Fundamentals and Experiences, pp. 79–93. Springer, Boston (2007)
5. Becker, J., et al.: Configurative Method Engineering – On the Applicability of Reference Modeling Mechanisms in Method Engineering. In: Proceedings of the 13th Americas Conference on Information Systems (AMCIS 2007), Keystone, pp. 1–12 (2007)
6. Braun, C., et al.: Method Construction – A Core Approach to Organizational Engineering. In: Haddad, H., et al. (eds.) Proceedings of the 20th Annual ACM Symposium on Applied Computing (SAC 2005), Santa Fe, New Mexico, USA, pp. 1295–1299 (2005)

7. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38, 275–280 (1996)
8. Brookes, C.H.P., et al.: Information Systems Design. Prentice-Hall, Sydney (1982)
9. Bucher, T., et al.: Situational Method Engineering – On the Differentiation of "Context" and "Project Type". In: Ralyté, J., et al. (eds.) Situational Method Engineering – Fundamentals and Experiences, pp. 33–48. Springer, Boston (2007)
10. Bucher, T., Winter, R.: Dissemination and Importance of the "Method" Artifact in the Context of Design Research for Information Systems. In: Vaishnavi, V., Baskerville, R. (eds.) Proceedings of the Third International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008), Atlanta, pp. 39–59 (2008)
11. Chmielewicz, K.: Forschungskonzeptionen der Wirtschaftswissenschaft. Schäffer-Poeschel, Stuttgart (1994)
12. Fettke, P., Loos, P.: Classification of reference models: a methodology and its application. Information Systems and e-Business Management 1, 35–53 (2003)
13. Fettke, P., Loos, P.: Multiperspective Evaluation of Reference Models – Towards a Framework. In: Jeusfeld, M.A., Pastor, Ó. (eds.) Proceedings of the Conceptual Modeling for Novel Application Domains Workshop (ER2003 Workshops ECOMO, IWCMQ, AOIS, and XSDM), Chicago, pp. 80–91 (2003)
14. Fettke, P., Loos, P.: Ontological Evaluation of Reference Models Using the Bunge-Wand-Weber Model. In: DeGross, J.I. (ed.) Proceedings of the Ninth Americas Conference on Information Systems, Tampa, pp. 2944–2955 (2003)
15. Fettke, P., Loos, P.: Der Beitrag der Referenzmodellierung zum Business Engineering. HMD – Praxis der Wirtschaftsinformatik 42, 18–26 (2005)
16. Gorla, N., Umanath, N.S.: On the Design of Optimal Compensation Structures for Outsourcing Software Development and Maintenance: An Agency Theory Perspective. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, pp. 646–662 (2006)
17. Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H.: Configurable Process Models: A Foundational Approach. In: Lehner, F., et al. (eds.) Proceedings of the Multikonferenz Wirtschaftsinformatik 2006 (MKWI 2006), Passau (2006)
18. Grabowski, H., Lossack, R.-S., El-Mejbri, E.-F.: Towards a Universal Design Theory. In: Kals, H., van Houten, F. (eds.) Integration of Process Knowledge Into Design Support Systems: Proceedings of the 1999 CIRP International Design Seminar, University of Twente, Enschede, The Netherlands, Dordrecht, March 24-26, 1999, pp. 49–56 (1999)
19. Green, P., Rosemann, M.: Integrated Process Modeling: An Ontological Evaluation. Information Systems 25, 73–87 (2000)
20. Hevner, A.R., et al.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
21. Karlsson, F., Ågerfalk, P.J.: Method configuration: adapting to situational characteristics while creating reusable assets. Information and Software Technology 46, 619–633 (2004)
22. Knackstedt, R.: Fachkonzeptionelle Referenzmodellierung einer Managementunterstützung mit quantitativen und qualitativen Daten – Methodische Konzepte zur Konstruktion und Anwendung, Doctoral Thesis, University of Münster, Münster (2004)
23. Knackstedt, R., Janiesch, C., Rieke, T.: Configuring Reference Models – An Integrated Approach for Transaction Processing and Decision Support. In: Manolopoulos, Y., et al. (eds.) Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS 2006), Paphos, pp. 135–143 (2006)

24. Kornyshova, E., Deneckère, R., Salinesi, C.: Method Chunks Selection by Multicriteria Techniques: an Extension of the Assembly-based Approach. In: Ralyté, J., et al. (eds.) Situational Method Engineering – Fundamentals and Experiences, pp. 64–78. Springer, Boston (2007)
25. Kuechler, W.L., Vaishnavi, V.K.: Theory Development in Design Science Research: Anatomy of a Research Project. In: Vaishnavi, V.K., Baskerville, R. (eds.) Proceedings of the Third International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008), Atlanta, pp. 1–15 (2008)
26. Kunene, K.N., Weistroffer, H.R.: Design of a Method to Integrate Knowledge Discovery Techniques with Prior Domain Knowledge for Better Decision Support. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 343–355. Claremont (2006)
27. Lehner, F.: Modelle und Modellierung in angewandter Informatik und Wirtschaftsinformatik oder wie ist die Wirklichkeit wirklich? Research Report No. 10, Chair of Business Information Science and Information Management, WHU – Otto Beisheim School of Management, Koblenz (1994)
28. Leppänen, M.: An Ontological Framework and a Methodical Skeleton for Method Engineering. Doctoral Thesis, University of Jyväskylä, Jyväskylä (2005)
29. Lossack, R.-S., Grabowski, H.: The Axiomatic Approach in the Universal Design Theory. In: Tate, D. (ed.) Proceedings of the First International Conference on Axiomatic Design (ICAD 2000), Cambridge, MA, pp. 203–210 (2000)
30. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
31. Mirbel, I., Ralyté, J.: Situational method engineering: combining assembly-based and roadmap-driven approaches. Requirements Engineering 11, 58–78 (2006)
32. Peffers, K., et al.: The Design Science Research Process: A Model for Producing and Presenting Information Systems Research. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 83–106. Claremont (2006)
33. Pfeiffer, D., Niehaves, B.: Evaluation of Conceptual Models – A Structuralist Approach. In: Bartmann, D., et al. (eds.) Proceedings of the 13th European Conference on Information Systems (ECIS 2005), Regensburg (2005)
34. Ralyté, J., Rolland, C.: An Approach for Method Reengineering. In: Hideko, S.K., et al. (eds.) ER 2001. LNCS, vol. 2224, pp. 471–484. Springer, Heidelberg (2001)
35. Rolland, C.: A Primer for Method Engineering. In: Proceedings of the Informatique des Organisations d'Information et de Décision (INFORSID). Toulouse (1997)
36. Schelp, J., Winter, R.: Method Engineering – Lessons Learned from Reference Modeling. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 555–575. Claremont (2006)
37. Schütte, R., Rotthowe, T.: The Guidelines of Modeling – An Approach to Enhance the Quality in Information Models. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 240–254. Springer, Heidelberg (1998)
38. Tatnall, A., Davey, B., McConville, D.: Information Systems – Design and Implementation. Data Publishing, Melbourne (1996)
39. Tolvanen, J.-P.: Incremental Method Engineering with Modeling Tools: Theoretical Principles and Empirical Evidence. Doctoral Thesis, University of Jyväskylä, Jyväskylä (1998)

40. van de Weerd, I., Brinkkemper, S.: Meta-Modeling for Situational Analysis and Design Methods. In: Syed, M.R., Syed, S.N. (eds.) Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, pp. 35–54. Idea Group, Hershey (2008)
41. van Slooten, K., Hodes, B.: Characterizing IS development projects. In: Brinkkemper, S., et al. (eds.) IFIP TC8 Working Conference on Method Engineering, pp. 29–44. Chapman & Hall, London (1996)
42. Venable, J.R.: The Role of Theory and Theorising in Design Science Research. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the 1st International Conference on Design Science in Information Systems and Technology (DESRIST 2006), pp. 1–18. Claremont (2006)
43. Vessey, I., Ramesh, V., Glass, R.L.: Research in Information Systems – An Empirical Study of Diversity in the Discipline and Its Journals. Journal of Management Information Systems 19, 129–174 (2002)
44. vom Brocke, J.: Referenzmodellierung: Gestaltung und Verteilung von Konstruktionsprozessen. Doctoral Thesis, University of Münster, Münster (2003)
45. von Brocke, J.: Design Principles for Reference Modelling. Reusing Information Models by Means of Aggregation, Specialisation, Instantiation, and Analogy. In: Fettke, P., Loos, P. (eds.) Reference Modelling for Business Systems Analysis, pp. 47–75. Idea Group, Hershey (2007)
46. von Brocke, J., Buddendick, C.: Reusable Conceptual Models – Requirements Based on the Design Science Research Paradigm. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 576–604. Claremont (2006)
47. Winter, R., Schelp, J.: Reference Modeling and Method Construction – A Design Science Perspective. In: Liebrock, L.M. (ed.) Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC 2006), pp. 1561–1562 (2006)
48. Zhao, H.: Selective Encryption for MPEG-4 FGS Videos. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 605–609. Claremont (2006)