

Contextual Method Engineering

Authors: Tobias Bucher, Mario Klesse
Chair: Prof. Dr. Robert Winter
Report No.: BE HSG/EIW/03
Date: November 8, 2006

University of St. Gallen (HSG)

Graduate School of Business, Economics, Law and Social Sciences

Institute of Information Management
Müller-Friedberg-Strasse 8
CH-9000 St. Gallen
Phone: + 41 (0) 71 224 3800
Fax: + 41 (0) 71 224 2189

Prof. Dr. A. Back
Prof. Dr. W. Brenner
Prof. Dr. H. Österle
Prof. Dr. R. Winter

Table of Contents

Abstract	3
1 Introduction	4
2 Related Work.....	5
2.1 Situational Method Configuration.....	5
2.2 Situational Method Composition	6
2.3 Research Gap.....	7
3 Situation and Context within Method Engineering.....	9
4 A Process for Contextual Method Engineering.....	12
5 Conclusion and Further Research	15
References	16

Abstract

Based on the experience that there cannot be a “one-size-fits-all” method, different situational method engineering approaches are examined in this paper. The result of the analysis shows that “situations” are conceptualized very imprecisely. Therefore, we propose to differentiate between situation and context in situational method engineering. Especially context is neglected in existing method engineering approaches. To close this gap, we enhance the existing engineering processes by adding three steps to facilitate the identification of context factors, enabling the engineering of contextual methods.

1 Introduction

The method engineering (ME) discipline is concerned with the processes of designing, constructing, and adapting generic artifacts such as models, methods, techniques, and tools aimed at the development of information systems or work systems [Brinkkemper 1996]. According to the definition of [Alter 2003, Alter 2006], an information system (IS) can be classified as a dedicated form of a work system. Therefore, we refer to the objects that are to be engineered or transformed by means of a method as “work systems” (WS) in the following.

A “method” is commonly understood as a “systematic way of acting that leads to revisable results” [Greiffenberg 2003], or as a “process that is built upon a set of rules, yielding solutions to a particular problem type” [Teubner 1999]. Similarly, [Becker et al. 2001] characterize a method as a tuple of a task and a set of rules: the former defines the problem type that is addressed by the method, and the latter describes prescribed actions aiming at an adequate solution to the problem.

In order to be applicable for WS development, methods need to be adapted to the specific characteristics of the so-called development situation or project situation. This approach is commonly referred to as “situational method engineering” [Harmsen 1997] and may be ascribed to Fiedler’s “contingency model” [Fiedler 1964]. According to this scientific theory, there is no “best way” of organizing or leading a company. On the contrary, there are various internal and external factors that influence organizational effectiveness, and therefore the organizational style must be contingent upon those factors. This theory was often transferred to WS development in the past [Schonberger 1980, Weill/Olson 1989, Arinze 1991] and apparently also to the field of method engineering.

The paper at hand is aimed at contributing to the method engineering discipline by proposing a new approach to situational method engineering that explicitly addresses the difference between “situation” and “context”. The remainder of the paper is structured as follows: In section 2, related work on situational method engineering is discussed in depth. Based on the conclusion that there is no generally accepted understanding of what is meant by the term “situation”, a model of situation and context within WS development method engineering is proposed in section 3. In section 4, a procedure model for contextual method engineering is sketched. Finally, implications and further research are discussed in section 5.

2 Related Work

Change constructions of generic artifacts such as models or methods always bear reference to an initial artifact (model, method, partial model, method fragment, method chunk) in contrast to which they are configured and/or composed with regard to the characteristics of a specific scenario or project type. In addition to the adaptation of generic artifacts, change constructions can as well follow the so-called compositional principle. According to this approach, new constructional results are established by assembling artifact fragments (partial models, method fragments, method chunks).

Correspondingly, vom Brocke [2003] differentiates between the two change construction techniques “configuration” and “aggregation”. The “configuration” technique follows the so-called adaptive principle, i.e. subsequent changes are explicitly allowed for and planned already at the moment of the initial construction of the artifact. On the other hand, the aggregation technique follows the compositional principle, permitting changeability that is, at least to a certain degree, almost unrestricted.

Following this systematization, we propose to divide change mechanisms and reuse concepts of the method engineering discipline into “situational method configuration” on the one hand and “situational method composition” on the other hand.

2.1 Situational Method Configuration

The distinguishing mark of situational method configuration is the adaptation of a so-called base method against the background of a specific development situation or project situation of WS development [Karlsson et al. 2001, Karlsson/Ågerfalk 2004]. According to Karlsson/Ågerfalk, the adaptation process of a generic method is organized in three distinct phases:

- *Defining Configuration Packages.* A Configuration Package (CP) represents the configuration of a base method with respect to the characteristics of one single, well-defined part of a development situation.
- *Combining CPs in Configuration Templates.* A Configuration Template (CT) represents the comprehensive configuration of a base method with respect to a vector of recurrent project characteristics mapped onto a development situation that comprehends several delimited characteristics. Thus, a CT is based on a specific combination of CPs.
- *Selecting a CT that is adequate for the project situation.* By identifying the characteristics of a project situation and matching them with the characteristics of a CT, an adequate configuration of the base method with respect to the project situation can be obtained.

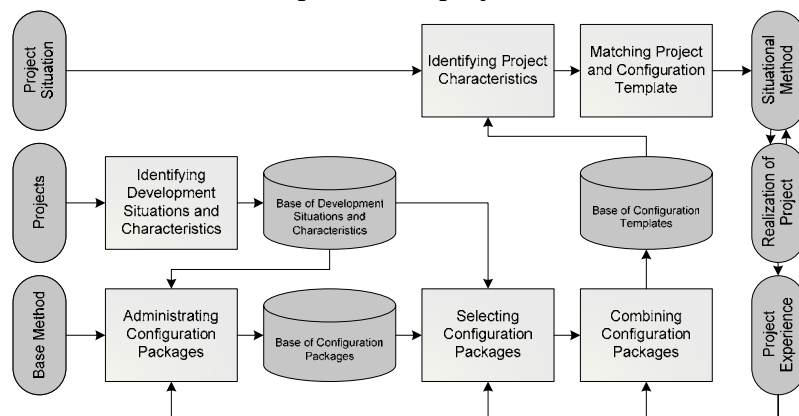


Fig. 1. Situational method configuration [Karlsson/Ågerfalk 2004]

The configuration process proposed by Karlsson and Ågerfalk [2004] (cf. figure 1) is characterized by its systematic structure and its intuitive comprehensibility. However, the authors do not provide a proper definition of what is actually meant by the terms “development situation” and “project situation” respectively. Karlsson and Ågerfalk merely point out that a development situation “is an abstraction of one or more existing or future software development projects with common characteristics”. However, the authors do not offer any guidance in identifying and/or defining those characteristics.

2.2 Situational Method Composition

The fundamental idea of situational method composition is the situational selection and orchestration of artifact fragments with respect to the specifics in a WS development situation. Unlike situational method configuration, the composition process is not aimed at configuring one single base method but at combining and aggregating several method chunks in order to establish new constructional results. This approach to situational method engineering (cf. figure 2) is widely-used and discussed in-depth in the scientific literature, e.g. [Brinkkemper 1996, Punter/Lemmen 1996, Roland/Prakash 1996, Harmsen 1997, Brinkkemper et al. 1998, Baumöl 2005].

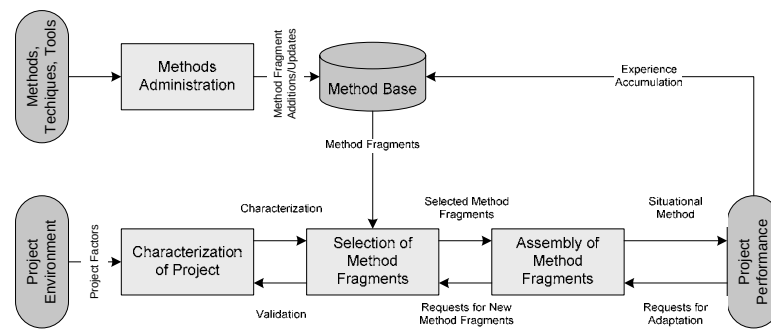


Fig. 2. Situational method composition [Brinkkemper 1996, Harmsen 1997]

Based on the seminal contributions of Brinkkemper and Harmsen, the composition process can as well be subdivided into three phases:

- *Identifying situational characteristics.* As a first step, it is necessary to identify situational characteristics that can be used both for characterizing specific development project types as well as artifacts and artifact fragments respectively.
- *Decomposing generic artifacts into artifact fragments.* In order to fill the method base, generic artifacts like methods, techniques, and tools need to be decomposed into artifact fragments. Furthermore, the artifact fragments and their interrelationships need to be described by use of the situational characteristics defined in the first phase of the process.
- *Composing artifact fragments into a situational method.* The actual composition of a situational method takes place in the third phase by choosing and orchestrating artifact fragments according to well-defined construction or composition principles in order to fit the situational characteristics of the development project.

Although even early contributions to situational method composition put significant emphasize on the necessity to identify situational characteristics [Brinkkemper 1996, Harmsen 1997, Brinkkemper et al. 1998], merely two articles offer some guidance regarding this requirement:

Punter and Lemmen [1996] propose to apply the MADIS modeling framework [Essink 1986, Essink 1988] for the characterization of the problem situation at the one side and the artifact fragments

on the other side. The underlying idea of MADIS is that the WS development process can be viewed at different levels of abstraction (object system modeling, conceptual IS modeling, data system modeling, implementation modeling). At each level, identical aspect domains (goal structure, environmental interaction, functional structure, entity structure, process structure, system dynamics, allocation aspect, realization aspect) have to be considered as part of the development process [Punter/Lemmen 1996].

According to Rolland and Prakash [1996], a development situation is characterized both by the problem domain and the subject area (cf. figure 3). Within the problem domain, the situational factors complexity (simple, moderate, complex) and risk (low, moderate, high) are evaluated for both the target domain and the project domain [Rolland/Prakash 1996]. With respect to the specification of the problem domain, the authors refer to [Franckson 1994] but fail to explicitly derive or state reasons for the choice of the situational factors.

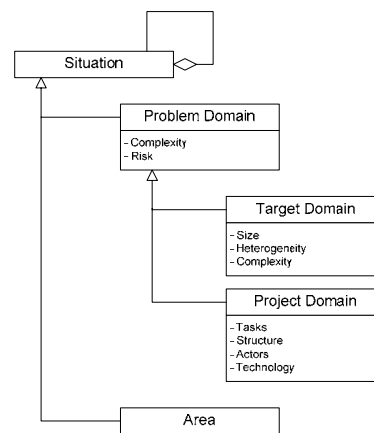


Fig. 3. Specification of a development situation [Rolland/Prakash 1996]

2.3 Research Gap

Irrespective of the preferred way of configuring (cf. section 2.1) or composing (cf. section 2.2) generic methods with regard to the characteristics of a specific scenario or project type, there is general agreement among all authors dealing with issues of situational method engineering that one needs to explain the characteristics of the relevant development situation and to adapt generic methods with respect to these situational characteristics. However, there is obvious dissent among the authors about what exactly is meant by the word “situational”. Table 1 gives an overview of some related concepts.

Author	Concept	Concept description
[Brinkkemper 1996, Harmsen 1997, Brinkkemper et al. 1998]	Project environment	Since every project is different, the project environment must be characterized according to a list of contingency factors.
[Punter/Lemmen 1996]	Project environment	The project environment is characterized by the problem situation and its contingency. Both the problem situation and the method fragments are described using the MADIS framework.
[Rolland/Prakash 1996]	Situation	The development situation is characterized by the problem domain and the subject area. Within the problem domain, the situational factors complexity and risk are evaluated for both the target domain and the project domain
[Karlsson et al. 2001, Karlsson/Ågerfalk 2004]	Project situation	The project situation describes the characteristics of a specific software development project.
	Development situation	A development situation is an abstraction of one or more existing or future software development projects with common characteristics.
[Baumöl 2005]	Reference context	Reference contexts are abstractions of project types that share common characteristics.

Table 1. Concepts used for specifying situational characteristics of project situations

Aside from the contributions of Punter and Lemmen as well as of Rolland and Prakash, current research in the field of situational method engineering offers no support at all in how to actually specify WS development situations.

Therefore, the paper at hand is aimed at bridging this gap by proposing a terminological differentiation between the concepts “situation” and “context” as well as by sketching a procedure model that incorporates the idea of context into the situational method construction processes.

3 Situation and Context within Method Engineering

All existing method concepts in the discipline of method engineering contain a procedure/activity model to accomplish the creation or transformation of a certain artifact [Braun et al. 2005], also referred to as “product” by some authors [Prakash 1999]. While for IS methods this artifact is mainly an information system [Truex/Avison 2003], the concept of a method is also applicable for engineering and transforming work systems [Baumöl 2005]. As stated in the introduction, we stick to the term “work system” [Alter 2003] to subsume any systems where methods are applicable on. These systems consist of one or more system elements that are to be engineered or changed by the method. In the following, we use the term “transformation” for any engineering or change of a work system.

Consequently, a method can be viewed as a systematic aid that helps transforming a work system from an initial state to a target state (cf. figure 4). In the following, we name the work system that is transformed by the method’s activities WS_S , its initial state S_A and its target state S_Z . WS_S consists of a set of system elements [Rüegg-Stürm 2002] that are transformed by the method.

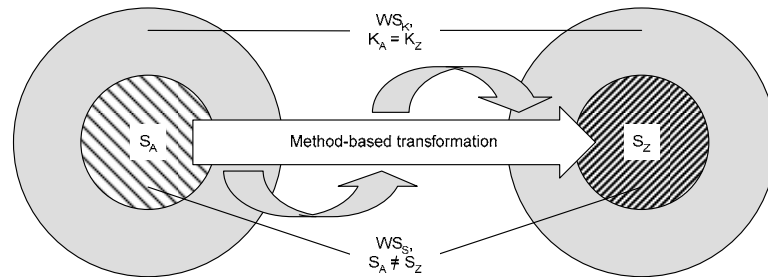


Fig. 4. Situation and context of business and information systems within method engineering

The tuple $(S_A; S_Z)$ is qualified as “situation” by some authors [Rolland/Prakash 1996, Karlsson/Ågerfalk 2004]. This means that a situation can be characterized by an initial state and a designated target state of a work system:

$$S_A := \text{initialStateOf}[WS_S] \quad (1)$$

$$S_Z := \text{targetStateOf}[WS_S] \quad (2)$$

$$\text{Situation} := (S_A; S_Z) \quad (3)$$

A more suitable designation for “situation” might be “project type” or “task type”. Nevertheless, we retain the term “situation” because it is part of the established terminology in method engineering. Table 2 shows two example situations that might come up in data warehouse development.

Method artifact WS_S	Situation description	Initial state S_A	Target state S_Z
Informational/ analytical IS landscape and its supporting IT processes (for elements see e.g. [Inmon 1997, Kachur 2000, Inmon 2002])	Data warehouse development, “green field” (initial development)	Elements are non-existent.	Data warehouse is implemented and filled with initial data, development, operation, and support processes are established.
	Data warehouse development by integrating independent data marts (consolidation)	Independent data marts are existent, maintained and are regularly refreshed with data, data between data marts is inconsistent, no data warehouse does exist.	Data warehouse is implemented and filled with initial data, development, operation, and support processes are established, former data marts are abolished, platforms are reused as far as possible.

Table 2. Examples of situations in method engineering

Situations influence effectiveness and efficiency of method application [Baumöl 2005]. This fact leads to the concept of situational method engineering where methods or method chunks are treated as situation-specific (cf. section 2). For each situation a suitable method has to be constructed either by recombining existing situation method chunks or by adapting an existent method to the situation. A way to systematically identify situations is presented in [Baumöl 2005].

Besides situation, there are other – environmental – contingency factors [e.g. Ives et al. 1980] that influence effectiveness and efficiency of method application. It is a matter of fact that each WS_S is part of a larger work system; e.g. an information system is part of an IS landscape, an IS landscape is part of a company, and a company is part of a business network. We refer to this larger work system as WS_G , with WS_S as a subset of WS_G . All elements that are not part of WS_S , but part of WS_G can be referred to as environmental work system of WS_S . This environment WS_K is outside the transformation scope of a method. It may consist of non-changeable system elements or of system elements that may not be changed. Although outside the method’s transformation scope, the state of these environmental work system elements may influence the applicability of transformation procedures or techniques, e.g. in form of restrictions. In the following, we refer to the state of this environment as “context”. The context is invariant during method application:

$$K_A := \text{initialStateOf}[WS_K] \quad (4)$$

$$K_Z := \text{targetStateOf}[WS_K] \quad (5)$$

$$\text{Context} := K_A = K_Z = K = \text{stateOf}(WS_K) = \text{stateOf}(WS_G \setminus WS_S) \quad (6)$$

Table 3 lists some exemplary context element states of a method and their potential impact on method applicability.

Method artifact WS_S	Context description	Possible impact
Informational/ analytical IS landscape and its supporting IT processes (for elements see e.g. [Inmon 1997, Kachur 2000, Inmon 2002])	Large company (>10000 employees) and large data warehouse	- Building a permanent organizational unit for data warehousing is strongly indicated to maintain the data warehouse [Winter/Meyer 2001].
	Medium company (1000 < no. of employees <=10000) and small data warehouse	- Building a permanent organizational unit for data warehousing is not cost-effective to maintain the data warehouse, instead roles should be integrated into existing business organization [Limacher 2005].

Table 3. Examples of contexts in method engineering

To summarize, both context and situation are relevant factors to be considered during method construction and application.

For our concept of situation, a development methodology to identify and structure project types is proposed in [Baumöl 2005]. Most existing ME approaches are designed to consider situation (cf. section 2).

For the concept of context, such a methodology is missing, and context is neglected by existing method engineering approaches focusing on product or project types (cf. section 2).

Therefore, in the next section we describe a procedure for context-adequate method construction that can be integrated into any existing situational method engineering methodology. Since the context may become very complex due to the theoretically unlimited size of the environment of WS_s , relevant factors have to be identified that describe the context. Afterwards, these context factors can be used as configuration parameters or as method metadata as it is proposed for situation descriptors in situational method engineering approaches [e.g. Rolland/Prakash 1996].

4 A Process for Contextual Method Engineering

The purpose of the process is to identify the context(s) in which a method component has to be valid, and to engineer a suitable method for these context settings. Since there are several method engineering approaches, we build our procedure around any epistemologically valid method engineering methodology fulfilling the following criteria:

- The method construction process results in a method or method component that is suitable for a given situation.
- The method construction process contains a validation of the constructed method component.
- The method construction process is iterative, i.e. the method can be further refined after construction.

All methodologies described in section 2 fulfill these criteria. Our process consists of four steps, incorporating the selected method construction methodology as last step (cf. figure 5). In the following, we describe each step and illustrate it with an example of designing a method in the field of data warehousing.

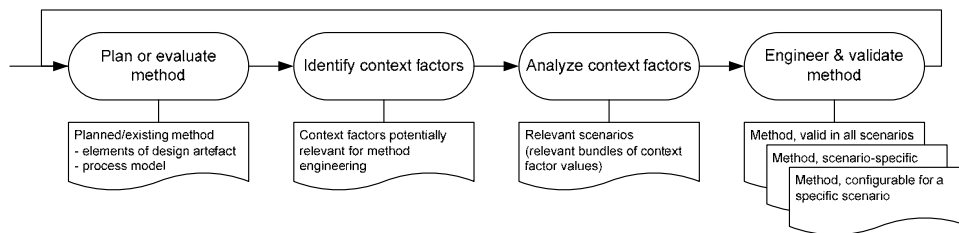


Fig. 5. Generic process for contextual method engineering

Step 1: Plan or evaluate method

As we worked out in the previous section, the context influences the appropriateness of method application as well as the design of the method's product/artifact. Therefore, as a prerequisite for identifying context factors, at least a rough idea about the method itself and the method's product has to exist. We distinguish between two initial situations of our process:

1. No method for a given situation exists yet: In this case, a method has to be built from scratch. If so, the procedure model and a product model of the planned method have to be outlined first.
2. A method for a given situation exists: In this case, the method has to be enhanced for contextual appropriateness. If so, the procedure model and the product model can be extracted from the existing method.

Example. We plan to develop a method for implementing an IT costing and charging process for data warehouse organizations. As basis we identify an existing method that is based on the IT Infrastructure Library (ITIL). This method is not context-specific, i.e. it claims to be suitable for all IT organizations. Its products are a cost model, IT services, and a charging model to be implemented in the IT organization.

Step 2: Identify context factors

For the procedure and product model extracted in step 1 context factors are identified. This step includes a systematic screening of the existing body of knowledge about the product and of existing procedures to design the product. This includes:

- Existing models, methods, procedure models and theories about the method's artifact,
- existing generic knowledge about procedure models, and
- experiences from practice projects or gained by observation.

The list of collected context factors may become very long. To reduce the list to potentially relevant context factors, the following criteria should be applied:

- There is empirical or theoretical evidence that the context factors have an impact on the method artifact and/or on the procedure to design the artifact.
- Context factors are invariant during method application. Otherwise, they are part of the method's product and are input for situational method engineering (step 4).

To compact the list, context factors should be classified and aggregated respectively systematized in a hierarchy. For each context factor, possible values and/or its range have to be specified.

Example. Screening the body of literature we identified three potentially relevant context factors that restrict the applicability of the ITIL-based method: The maturity stage of the IT organization and systems [VanLengen/Morgan 1993], the coordination form between IT and business organization [McKinnon/Kallman 1987], and the organizational positioning of the data warehouse service provider [Winter/Meyer 2001] affecting the IT services offered. Maturity can be assessed by CMMI stages of the IT processes [Ahern et al. 2003]. Coordination form can be expressed with the responsibility center concept (cost center, profit center, investment center) [Earl 1989]. Organizational positioning can be measured by the activities that are within responsibility of the data warehouse organization and of the business organization [Klesse/Winter 2006].

Step 3: Analyze context factors

In reality, not all context factor values do exist in any combination. Consequently, a method does not have to address all possible permutations of context factor values but only those combinations occurring with a certain frequency in practice. To extract these combinations, an exploratory empirical investigation can be conducted. This investigation should survey all potentially relevant context factors and their values in the population for which the method should be applicable. Relevant context factor value combinations can be extracted from the collected data by factor analysis and cluster analysis techniques [Tabachnick/Fidell 2006]. The result of this analysis should be "scenarios" bundling context factor values that are common in reality.

Example. The analysis of the context factors led to the following results: Data warehouse organizations have reached a medium to high level of maturity today [Chamoni/Gluchowski 2004]. Most data warehouse organizations are responsible for costs only [Klesse 2005] and positioned as "data warehouse competence centers" or "business service providers" according to their activity profile [Klesse/Winter 2006].

Step 4: Engineer and validate method

For these scenarios, one or more method fragments have to be constructed in step four. Depending on the method engineering approach, the resulting methods can be of the following types:

- Method, configurable for scenario: This type of method can be applied to a real-world-context by analyzing the context factors and configuring it for this context.
- Method, scenario-specific: This type of method is only applicable in one or more scenarios it was developed for.
- Method, scenario-independent: This type of method is applicable in all scenarios identified in step 3.

Regardless of the method type developed, the method has to be validated by applying it in the scenario it is claimed to be valid for.

Example. Since the most probable scenario of a data warehouse service provider is a mature “data warehouse competence center” responsible for costs, a method has to be developed and validated for this scenario only.

5 Conclusion and Further Research

Our analysis of existing method engineering approaches showed that nearly all approaches claim to incorporate situation or context factors. Nevertheless, the existing engineering methodologies do not describe what these situational or context factors comprise of and how to identify them.

In our paper, we presented a new approach to situational method engineering. Based on the conclusion that situation and context are different determinants of method applicability, we proposed a procedure model that incorporates context into (situational) method construction processes. Our procedure model guides the identification of relevant context factors, examines their occurrences in practice, and classifies them into scenarios. For these scenarios, suitable methods can be constructed that might be more appropriate than generic methods focusing solely on situations and not incorporating context factors.

Based on the initial work presented in this paper, three broad categories of research opportunities exist.

First, our process for identifying relevant context scenarios can be transferred to the identification of situations. This would enable a more systematic classification of situations. Using a systematic approach for identifying contexts and situations, common “reference” contexts and situations could be collected. With evidence about their frequency of occurrence in practice, information systems discipline could concentrate on developing situational and contextual methods for the most common scenarios and project types.

Second, our process could be extended to cover epistemologically valid method construction processes by evaluating and combining existing method engineering approaches described in section 2. Developing a reference process or even a method for method construction itself would help method engineers to systematically design new methods.

Third, a reference information model of method chunks and their context and situations could be designed. This would enable storing and managing situational/contextual method elements in an electronic method base, and therefore enhance reusability of method chunks.

References

[Ahern et al. 2003]

Ahern, D. M., Clouse, A., Turner, R., CMMI Distilled - A Practical Introduction to Integrated Process Improvement, 2, Addison Wesley, 2003

[Alter 2003]

Alter, S., 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, Nr. 23, S. 366-395

[Alter 2006]

Alter, S., Work Systems and IT Artifacts - Does the Definition Matter?, in: Communications of the Association for Information Systems, 17, 2006, Nr. 14, S. 299-313

[Arinze 1991]

Arinze, B., A Contingency Model of DSS Development Methodology, in: Journal of Management Information Systems, 8, 1991, Nr. 1, S. 149-166

[Baumöl 2005]

Baumöl, U., Strategic Agility Through Situational Method Construction, in: Reichwald, R.H., Anne Sigismund (Hrsg.), Proceedings of the European Academy of Management Annual Conference (EURAM2005), 2005, S.

[Becker et al. 2001]

Becker, J., Knackstedt, R., Holten, R., Hansmann, H., Neumann, S., Konstruktion von Methodiken: Vorschläge für eine begriffliche Grundlegung und domänenspezifische Anwendungsbeispiele, 77, Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik, Münster 2001

[Braun et al. 2005]

Braun, C., Wortmann, F., Hafner, M., Winter, R., Method Construction - A Core Approach to Organizational Engineering, in: Haddad, H., Liebrock, L.M., Omicini, A., Wainwright, R.L. (Hrsg.), Proceedings of the 20th Annual ACM Symposium on Applied Computing (SAC 2005), ACM, Santa Fe, New Mexico 2005, S. 1295-1299

[Brinkkemper 1996]

Brinkkemper, S., Method Engineering: Engineering of Information Systems Development Methods and Tools, in: Information and Software Technology, 38, 1996, Nr. S. 275-280

[Brinkkemper et al. 1998]

Brinkkemper, S., Saeki, M., Harmsen, F., Assembly Techniques for Method Engineering, in: (Hrsg.), Advanced Information Systems Engineering: 10th International Conference, CAiSE'98, Pisa, Italy, June 1998, Springer, Berlin 1998, S. 381-400

[Chamoni/Gluchowski 2004]

Chamoni, P., Gluchowski, P., Integrationstrends bei Business-Intelligence-Systemen: Empirische Untersuchung auf Basis des Business Intelligence Maturity Model, in: Wirtschaftsinformatik, 46, 2004, Nr. 2, S. 119-128

[Earl 1989]

Earl, M. J., Management Strategies for Information Technologies, Prentice Hall, New York 1989

[Essink 1986]

Essink, L. J. B., A Modelling Approach to Information System Development, in: Olle, T.W., Sol, H.G., Stuart, A.A.V. (Hrsg.), Information Systems Design Methodologies: Improving the Practice, North-Holland, Amsterdam 1986, S.

[Essink 1988]

Essink, L. J. B., A Conceptual Framework for Information Systems Development Methodologies, in: Bullinger, H.J. (Hrsg.), Eurinfo '88, Proceedings of the First European Conference on Information Technology for Organizational Systems, North-Holland, Amsterdam 1988, S.

[Fiedler 1964]

Fiedler, F. E., A Contingency Model of Leadership Effectiveness, in: Advances in Experimental Social Psychology, 1, 1964, Nr. S. 149-190

[Franckson 1994]

Franckson, M., The Euromethod Deliverable Model and Its Contribution to the Objectives of Euromethod, in: Verrijn-Stuart, Olle (Hrsg.), Proc. IFIP-TC8 Int. Conf. on Methods and Tools for the Information Systems Life Cycle, North-Holland 1994, S. 131-149

[Greiffenberg 2003]

Greiffenberg, S., Methoden als Theorien der Wirtschaftsinformatik, in: Uhr, W., Esswein, W., Schoop, E. (Hrsg.), Wirtschaftsinformatik 2003, Band 2, 2003, S. 947-968

[Harmsen 1997]

Harmsen, F., Situational Method Engineering, Moret Ernst & Young Management Consultants, Utrecht 1997

[Inmon 2002]

Inmon, W. H., Building the Data Warehouse, 3, Wiley, New York 2002

[Inmon 1997]

Inmon, W. H. W., J. D. & Glassey, Katherine L., Managing the Data Warehouse, Wiley, New York 1997

[Ives et al. 1980]

Ives, B., Hamilton, S., Davis, G. B., A Framework for Research in Computer-Based Management Information Systems, in: Management Science, 26, 1980, Nr. 9, S. 910-934

[Kachur 2000]

Kachur, R., Data Warehouse Management Handbook, Prentice Hall, USA 2000

[Karlsson/Ågerfalk 2004]

Karlsson, F., Ågerfalk, P. J., Method Configuration: Adapting to Situational Characteristics while Creating Reusable Assets, in: Information and Software Technology, 46, 2004, Nr. 9, S. 619-633

[Karlsson et al. 2001]

Karlsson, F., Ågerfalk, P. J., Hjalmarsson, A., Method Configuration with Development Tracks and Generic Project Types, in: (Hrsg.), Proceedings of the 6th CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in System Analysis and Design, 2001, S.

[Klesse 2005]

Klesse, M., Leistungsverrechnung im Data Warehousing - Ergebnisse einer empirischen Studie, Institut für Wirtschaftsinformatik der Universität St. Gallen, St. Gallen 2005

[Klesse/Winter 2006]

Klesse, M., Winter, R., Organizational Forms of Data Warehousing: An Explorative Analysis, Institute of Information Management, University of St. Gallen, St. Gallen 2006

[Limacher 2005]

Limacher, C., Organisationskonzept und Erfolgsfaktoren für den Betrieb und die Weiterentwicklung einer Data Warehouse Lösung am Beispiel des Universitätsspitals Zürich, Unveröffentlichte Bachelorarbeit, Universität St. Gallen, St. Gallen 2005

[McKinnon/Kallman 1987]

McKinnon, W. P., Kallman, E. A., Mapping Chargeback Systems to Organizational Environments, in: MIS Quarterly, 11, 1987, Nr. 1, S. 5-20

[Prakash 1999]

Prakash, N., On Method Statics and Dynamics, in: Information Systems Journal, 24, 1999, Nr. 8, S. 613-637

[Punter/Lemmen 1996]

Punter, T., Lemmen, K., The MEMA-Model: Towards a New Approach for Method Engineering, in: Information and Software Technology, 38, 1996, Nr. 4, S. 295-305

[Rolland/Prakash 1996]

Rolland, C., Prakash, N., A Proposal for Context-Specific Method Engineering, in: Brinkemper, S., Lytinnen, K., Welke, R.J. (Hrsg.), Method Engineering, Principles of Method Construction and Tool Support, Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering, Springer, Berlin 1996, S. 191-207

[Rüegg-Stürm 2002]

Rüegg-Stürm, J., Das neue St. Galler Management-Modell: Grundkategorien einer integrierten Managementlehre, Paul Haupt, Bern 2002

[Schonberger 1980]

Schonberger, R. J., MIS Design: A Contingency Approach, in: MIS Quarterly, 4, 1980, Nr. 1, S. 13-20

[Tabachnick/Fidell 2006]

Tabachnick, B. G., Fidell, L. S., Using Multivariate Statistics, 5, Allyn & Bacon, 2006

[Teubner 1999]

Teubner, R. A., Organisations- und Informationssystemgestaltung - Theoretische Grundlagen und integrierte Methoden, DUV/Gabler, Wiesbaden 1999

[Truex/Avison 2003]

Truex, D., Avison, D., Method Engineering - Reflections on the Past and Ways Forward, in: (Hrsg.), Ninth Americas Conference on Information Systems, 2003, S. 508-514

[VanLengen/Morgan 1993]

VanLengen, C. A., Morgan, J. N., Chargeback and Maturity of IS Use, in: Information & Management, 25, 1993, Nr. 3, S. 155-163

[vom Brocke 2003]

vom Brocke, J., Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen, Logos, Berlin 2003

[Weill/Olson 1989]

Weill, P., Olson, M. H., An Assessment of the Contingency Theory of Management Information Systems, in: Journal of Management Information Systems, 6, 1989, Nr. 1, S. 59-85

[Winter/Meyer 2001]

Winter, R., Meyer, M., Organization Of Data Warehousing In Large Service Companies: A Matrix Approach Based On Data Ownership And Competence Centers, in: (Hrsg.), Proceedings of the Seventh Americas Conference on Information Systems (AMCIS 2001), 2001, S. 322-328