Situational Method Engineering

On the Differentiation of "Context" and "Project Type"

Tobias Bucher, Mario Klesse, Stephan Kurpjuweit, Robert Winter
Institute of Information Management, University of St. Gallen
Mueller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
{tobias.bucher, mario.klesse, stephan.kurpjuweit, robert.winter}@unisg.ch
http://www.iwi.unisg.ch

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 "context" and "project type" in situational method engineering. Especially context is neglected in existing method engineering approaches. To close this gap, we enhance existing method engineering processes by adding three steps to facilitate the identification of context factors and project type factors, enabling the engineering of both contextual and project type-specific methods. Furthermore, we propose a set of extensions to the method engineering meta model that allow the method engineer to differentiate between "context" and "project type" in describing situational methods.

1 Introduction

Since its first theoretical foundation almost four decades ago, the "sciences of the artificial" [1] have evolved into a key research paradigm in the study of information systems development that is commonly referred to as "design research" [2, 3, 4]. Unlike research in behavioral or natural sciences, design research is not aimed at explaining the behavior of a system that is observable to the researcher but rather at creating solutions to specific problems of practical relevance. Typical outputs produced by design research are representational constructs (e.g. ontologies), models (e.g. architecture models, process models), methods, and instantiations (e.g. prototypes, reference models) [2, 3].

This paper aims at contributing to the state-of-the-art in a particular subset of design research, namely method engineering. Traditionally, method engineering (ME) is concerned with the processes of designing, constructing, and adapting methods

Please use the following format when citing this chapter:

Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R., 2007, in IFIP International Federation for Information Processing, Volume 244, Situational Method Engineering: Fundamentals and Experiences, eds. Ralyté, J., Brinkkemper, S., Henderson-Sellers B., (Boston Springer), pp. 33-48.

directed at the development of information systems [5]. According to the definition of Alter, an information system (IS) can be understood as a specific subtype of a work system [6, 7]. 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.

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" [5].

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" [8, 9, 10] and may be ascribed to the so-called "contingency model" proposed by Fiedler [11]. According to this scientific theory, there is no "best way" of organizing or leading an organization. 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 [12, 13, 14] and apparently also to the ME field.

The paper at hand is aimed at contributing to the ME discipline by proposing a new approach to situational ME that explicitly addresses the difference between "context" and "project type". The remainder of this paper is therefore structured as follows: In section 2, related work on situational ME is discussed. Based on the conclusion that there is no generally accepted understanding of what is meant by the term "situation", a model of context and project type within WS development ME (cf. [15]) is proposed in section 3. Section 4 is dedicated to the discussion of extensions to the ME meta model that has been proposed by Heym [16] and Gutzwiller [17]. The extensions recommended in this paper allow for the differentiation between "context" and "project type" when describing situational methods. In section 5, a procedure model for the engineering of contextual and project-specific methods (cf. [15]) is sketched. Implications and further research opportunities are discussed in the concluding section 6.

2 Discussion of 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.

Correspondingly, vom Brocke differentiates between the two modification techniques "configuration" and "aggregation" [18]. 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 subsequent changeability that is, at least to a certain degree, almost unrestricted.

Following this systematization, we propose to divide adaptation mechanisms of the ME discipline into “situational method configuration” on the one hand and “situational method composition” on the other hand (cf. [18]).

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 [19, 20]. According to Karlsson and 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.

The configuration process proposed by Karlsson and Ågerfalk [19] is characterized by its systematic structure and its intuitive comprehensibility [19, 20]. However, the authors do not provide a proper definition of what is actually meant by the terms “development situation” and “project situation” respectively (cf. table 1). They merely point out that a development situation “is an abstraction of one or more existing or future software development projects with common characteristics” [19] but fail to offer any guidance in identifying and/or defining those characteristics.

2.2 Situational Method Composition

The fundamental idea of situational method composition is the selection and orchestration of artifact fragments with respect to the specifics of 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 ME is widely-used and discussed in-depth in the scientific literature [5, 8, 10, 21, 22, 23, 24, 25].

Based on the seminal contributions of Brinkkemper [5] and Harmsen [8], the composition process can as well be subdivided into three phases:

- *Identifying situational characteristics.* Those characteristics can be used for characterizing specific development project types as well as artifacts and artifact fragments.

- *Decomposing generic artifacts into artifact fragments.* In order to fill the method base, generic artifacts need to be decomposed into artifact fragments. Furthermore, the artifact fragments and their interrelationships need to be described by use of the situational characteristics identified afore.
- *Composing artifact fragments into a situational method.* The actual composition of a situational method takes place 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 [5, 8, 9, 22], merely three articles offer some guidance regarding this requirement:

Punter and Lemmen [23] propose to apply the MADIS modeling framework [26, 27] 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 [23].

According to Rolland and Prakash [25], a development situation is characterized both by the problem domain and the subject area. 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 [25]. With respect to the specification of the problem domain, the authors refer to Franckson [28] but fail to explicitly derive or state reasons for the choice of the situational factors.

Van Slootes and Hodes [10] propose a list of 17 contingency factors whose values (ranging between low and high) influence the project approach, i.e. the specific method that has been adapted to fit the project context. The list of contingency factors comprises characteristics that are primarily external to the method application, i.e. they describe the environment to which the method is adapted and in which it is deployed.

2.3 Concepts Used for Specifying Situational Characteristics

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 ME 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 also obvious dissent among the authors about what exactly is meant by the term “situational”. Table 1 gives an overview of some related concepts.

Aside from the contributions of Punter and Lemmen [23], Rolland and Prakash [25], and van Slooten and Hodes [10], current research in the field of situational ME offers no support at all in how to actually specify WS development situations. Therefore, this paper is aimed at bridging the gap by proposing a terminological differentiation between the concepts "context" and "project type" as well as by sketching a procedure model that incorporates the idea of differentiating between context and project type into the situational method construction processes (cf. [15]).

Table 1. Concepts Used for Specifying Situational Characteristics

Author	Concept	Concept Description
Baumoel [21]	Reference Context	Reference contexts are abstractions of project types that share common characteristics.
Brinkkemper [5], Brinkkemper et al. [22], Harmsen [8]	Project Environment	Since every project is different, the project environment must be characterized according to a list of contingency factors.
Karlssoon and Ågerfalk [19], Karlsson et al. [20]	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.
Punter and Lemmen [23]	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 and Prakash [25]	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.
van Slooten and Hodes [10]	Project Context	The project context is made up of contingency factors that affect the project approach. Project context is external to method application.
	Project Approach	The project approach is the result of the configuration process of methods/method components.

3 "Situation" as Combination of "Context" and "Project Type"

All existing method concepts in the ME discipline comprise a procedure/activity model to accomplish the creation or transformation of a certain artifact [29], also referred to as "product" by some authors [30]. While for IS development methods this artifact is usually an information system [31], the concept of a method is also applicable for engineering and transformation of work systems [21]. As stated in the introduction, we stick to the term "work system" [6, 7] to subsume all systems which can be constructed by using methods. These systems comprise 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 WS.

Consequently, a method can be viewed as a systematic aid that helps transforming a WS from an initial state to a target state (cf. figure 1). In the following, we designate the WS that is transformed by the method's activities WS_S, its initial state S_A, and its target state S_Z. WS_S comprises a set of system elements [32] that are transformed by the method.

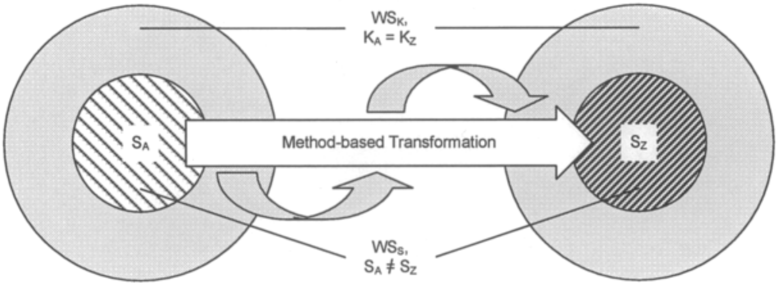


Fig. 1. Context and Project Type of Work Systems within ME [15]

The tuple (S_A, S_Z) is qualified as "situation" by some authors [19, 20, 25]. A more suitable designation for "situation" might be "project type" or "task type". A project type can be characterized by an initial WS state and a designated WS target state:

$$S_A := \text{initialStateOf}[WS_S] \tag{1}$$
$$S_Z := \text{targetStateOf}[WS_S] \tag{2}$$
$$\text{Project Type} := (S_A;\ S_Z) \tag{3}$$

Table 2 exhibits two exemplary project types that occur in data warehouse (DWH) development.

Table 2. Exemplary ME Project Types

Method Artifact WS_S	Project Type	Initial State S_A	Target State S_Z
Informational/analytical IS landscape and its supporting IT processes (for elements see e.g. [33, 34, 35])	"Green field" approach to DWH development (initial development)	Elements are non-existent	Data warehouse is implemented and filled with initial data; Development, operation, and support processes are established
	"Consolidation" approach to DWH development (by integrating existing, independent data marts)	Independent data marts are existent, maintained, and are regularly refreshed with data; Data between data marts is inconsistent; DWH is non-existent	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

The project type has significant impact on the effectiveness and efficiency of method application [21]. This fact leads to the concept of situational ME where methods or method chunks are treated as situation-specific (cf. section 2). For each project type, a suitable method has to be constructed either by recombining existing method chunks or by adapting an existent method to the respective project type. A way to systematically identify project types is presented by Baumoel [21].

Besides the project type, there are other – environmental – contingency factors (e.g. [36]) that also have significant impact on the effectiveness and efficiency of method application. It is a matter of fact that each WS_S is part of a larger WS; 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 of the transformation scope of a method. It may comprise non-transformable system elements. Although it is out of the method's transformation scope, the state of these environmental WS 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:

$$K_A := \text{initialStateOf}[WS_K] \tag{4}$$

$$K_Z := \text{targetStateOf}[WS_K] \tag{5}$$

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

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

Table 3. Exemplary ME Contexts

Method Artifact WS_S	**Context Description**	**Possible Impact**
Informational/analytical IS landscape and its supporting IT processes (for elements see e.g. [33, 34, 35])	Large company (more than 10'000 employees) and large DWH	Building a permanent organizational unit for data warehousing is strongly indicated to maintain the data warehouse [37]
	Medium company (number of employees between 1'000 and 10'000) and small DWH	Building a permanent organizational unit for data warehousing is not cost-effective to maintain the data warehouse. Instead roles should be integrated into the existing business organization [38]

In summary, both context and project type are relevant factors to be considered during method construction and application.

For our understanding of project type, a development methodology to identify and structure project types has been proposed by Baumoel [21]. Most existing ME approaches are designed to consider the project type (cf. section 2). For the concept of context, however, such a methodology is missing, and context is neglected in existing ME approaches. Therefore, we describe a three-step procedure for the engineering of both contextual and project type-specific methods in the section 5. This procedure can be integrated into any existing situational ME methodology (cf. step 4 of the process outlined in section 5). Since the context and project type may become very complex due to the theoretically unlimited size of the work system WS_S itself as well as of the environment of WS_S, relevant factors have to be identified that describe context(s) and project type(s) at the best possible rate. Afterwards, these factors can be used as configuration parameters or as method metadata as it is proposed for situation descriptors in situational ME approaches [25]. Before turning to the process, however, necessary extensions to the ME meta model are introduced that allow for the differentiation between context and project type in the description of situational methods.

4 Extensions to the Method Engineering Meta Model

Based on a review of different approaches to method construction and method implementation, Heym [16] and Gutzwiller [17] identified five constituent elements of a method: design activities, documents specifying design results, roles, techniques, and the information model of the method. By analyzing a total of twelve scientific contributions to the ME body of literature, Braun et al. [29] validated this set of elements that can be used for the description of generic methods. Therefore, it is reasonable to conclude that these five elements of work system design methods represent a "core" meta model.

As we have outlined in the paper at hand, this core is incomplete and insufficient for the adequate characterization of situational methods. We therefore propose to extend the ME core meta model by adding the three concepts "adaptation mechanism", "context", and "project type". In accordance with situational ME literature and with the ideas presented in the paper at hand, we regard "situation" as combination of context and project type. Any tuple of context characteristics and project type characteristics is referred to as situation. Furthermore, we introduce the "method fragment" as connecting factor for the adaptation mechanism element. According to Brinkkemper's definition that has been stated in section 1, a method provides goal-oriented instructions and recommendations [5]. Design activities describe the tasks that have to be executed (what?), and techniques specify possible ways in which the results can be achieved (how?). Therefore, we denote the combination of these two core elements as method fragment.

Our interpretation of the method fragment concept is largely in accordance with the so-called general method fragment meta model that has been derived by Cossentino et al. [39]. According to their research, a method fragment consists of the elements design activity, technique (referred to as "guidance"), role (referred to as "actor"), and design result (referred to as "artifact"). In our understanding, design activities and techniques are intrinsically tied to design results. Therefore, we dissent from Cossentino et al. solely with respect to the inclusion of the role element. In contrast to their perception, we do not explicitly call for the inclusion of roles. However, a role description may be attached to a design activity where required. The extended ME meta model is depicted in figure 2. Our extensions are highlighted by use of shaded element boxes and bold connectors.

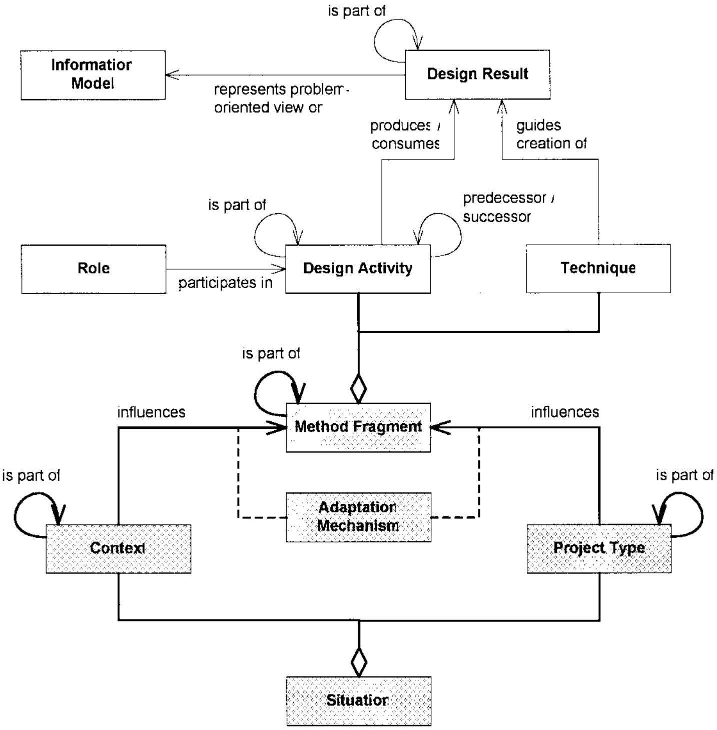


Fig. 2. Extended Method Engineering Meta Model

5 A Process for Engineering Contextual and Project Type-specific Methods

The purpose of the process proposed in this section is to identify the situations, i.e. tuples of context and project type, in which a method (or a method fragment) has to be valid, and to engineer a method that is suitable for these situations. The procedure can be combined with any ME methodology (cf. e.g. section 2). It consists of four steps, incorporating the selected method construction methodology as fourth step (cf. figure 3). In the following, we describe each step and illustrate it with an example of designing a method in the field of data warehousing.

Step 1: Plan or Evaluate Method

As we have shown in section 3, both context and project type influence the appropriateness of method application as well as the design of the method's product/artifact. Therefore, as a prerequisite for identifying factors that describe the context(s) and the project type(s), 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:

- *No method for a given situation exists yet.* Thus, a method has to be built from scratch. To this end, a procedure model and a product model of the method have to be outlined first.

- *A method for a given situation exists.* The method has to be enhanced for situational appropriateness. Both procedure and 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 DWH organizations. As basis we identify an existing method that is based on the IT Infrastructure Library (ITIL, cf. [40]). This method is not context-specific, i.e. it claims to be suitable for all IT organizations. The method's products are a cost model, IT services, and a charging model. Consequently, different project types such as "costing only" or "costing and charging" should be taken into consideration.

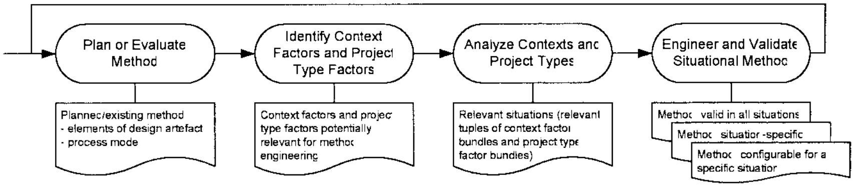

Plan or Evaluate Method
Identify Context Factors and Projec Type Factors
Analyze Contexts anc Project Types
Engineer and Validate Situational Methoc
Plannec/existing method
- elements of design artefact
- process mode
Context factors and projec type factors potentially relevant for methoc engineering
Relevant situations (relevant tuples of context factor bundles and project type factor bundles)
Methoc valid in all situations
Methoc situatior -specific
Methoc configurable for a specific situatior

Fig. 3. Generic Process for Engineering Contextual and Project Type-specific Methods (adapted from [15])

Step 2: Identify Context Factors and Project Type Factors

For the procedure and product model extracted in step 1, context factors and project type 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 products and includes:

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

The list of factors describing context and project type may become very long. To reduce it to potentially relevant factors, the following criteria should be applied:

- There is empirical or theoretical evidence that the factors have an impact on the method artifact and/or on the procedure to design the artifact.
- Context factors are invariant during method application.
- Project type factors change their values/characteristics during method application, i.e. they are part of the method's product.

To compact the list, both context factors and project type factors should be classified and aggregated/systematized into a hierarchy. For each factor, possible values and/or the values' range have to be specified.

At this juncture, techniques that are commonly used in behavioral and natural sciences are applied to the design research process in order to support the construction of design science artifacts. As pointed out by March and Smith [3] as well as by Cao et al. [41], interaction between different research paradigms, their methods, techniques, and activities are important since they complement each other in creating solutions to specific problems that are observable in the entrepreneurial world. The combination or "triangulation" (cf. [42]) of different methods and techniques will eventually lead to more enlightening and relevant research results and construction outcomes.

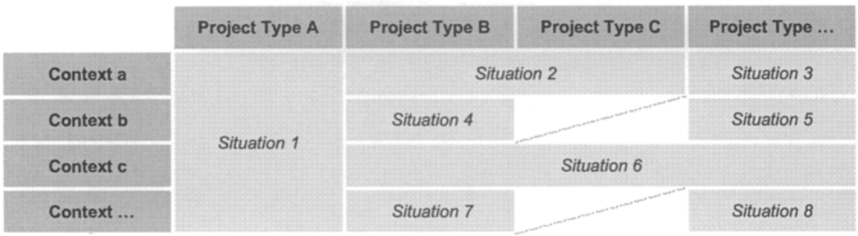
However, it is of particular importance to note that not all relevant context factors and/or project type factors are necessarily quantifiable in a trivial way. This is especially true if it comes to factors assessing organizational structures, complex social environments [43], workplace culture, or power. In this case, measurement constructs must be elaborated that can be used as auxiliary means for attributing numerical values to variables that would otherwise be non-quantifiable. This can be accomplished with the help of structural equation modeling (cf. e.g. [44]). A structural equation model (SEM) consists of a set of hypothetical constructs (so-called latent variables), a set of exogenous variables, and a set of endogenous variables. The model components are linked with each other by "directed" (i.e. causal) relationships. Ultimate goal of SEM is the prediction of interrelationships between endogenous, non-quantifiable variables through observation of exogenous, quantifiable variables.

Example. By screening the body of literature, two potentially relevant context factors and one project type factor restricting the applicability of the ITIL-based method can be found: the maturity stage of the IT organization and systems [45] and the organizational positioning of the DWH service provider [37] influence the context, and the coordination form between IT and business organization [46] influences the project type. Maturity can be assessed by CMMI stages of the IT processes [47]. Organizational positioning can be measured by the activities that are within responsibility of the DWH organization and of the business organization [48]. The coordination form can be expressed with the responsibility center concept (cost center, profit center, investment center) [49].

Step 3: Analyze Contexts and Project Types

In reality, not all context factor values and not all project type factor values do exist in any combination. Consequently, a method does not have to address all possible permutations of context factor values and project type 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 factors and their values in the population for which the method should be applicable. Relevant context factor value combinations and project type factor value combinations can be extracted by factor analysis and cluster analysis techniques (cf. e.g. [50]) – yet another use case of integrating techniques from behavioral and natural sciences into the design research process. The results of these analyses are contexts bundling context factor values that are common in reality and project types that bundle project type factor values that occur in a multiplicity of settings.

The identified context(s) and project type(s) can be summarized in matrix format (cf. figure 4). As already stated before, a tuple of context(s) and project type(s) is referred to as situation. A situation can relate to one or multiple contexts as well as to one or multiple project types. Moreover, it is important to note that certain combinations of context and project type might not exist in reality. Therefore, the complete enumeration of all situations might not lead to success at any rate.



	Project Type A	Project Type B	Project Type C	Project Type ...
Context a	*Situation 1*	*Situation 2*		*Situation 3*
Context b		*Situation 4*		*Situation 5*
Context c		*Situation 6*		
Context ...		*Situation 7*		*Situation 8*

Fig. 4. The Context vs. Project Type Matrix (exemplary illustration)

Example. The analysis of the context factors and project type factors led to the following results: DWH organizations have reached a medium to high level of maturity today [51]. Most DWH organizations are responsible for costs only [52] and positioned as “DWH competence centers” or “business service providers” according to their activity profile [48]. Thus, the matrix shows two contexts (“medium to high maturity, DWH competence centers” and “medium to high maturity, business service provider”) and one project type (“costing only”).

Step 4: Engineer and Validate Situational Method

For these situations, either a monolithic method or multiple method fragments that can be composed into a situational method have to be constructed. Depending on the ME approach (cf. section 2), the resulting methods can be of the following types:

- *Method, configurable for a specific situation.* This type of method can be applied to a real world setting by analyzing the context factors and project type factors and by configuring it with respect to the situation at hand. Consequently, step 4 of our process can correspond either to a method configuration process as proposed by Karlsson and Ågerfalk [19, 20] (cf. section 2.1) or to a method composition process as proposed by Brinkkemper [5] and Harmsen [8] (cf. section 2.2). In particular the first two phases of the configuration process (“defining configuration packages” and “combining configuration packages in configuration templates”) and the first phase of the composition process (“identifying situational characteristics”) are supported by our process. By identifying and analyzing relevant contexts and project types, the situational appropriateness of configuration packages, configuration templates, characterizations of specific development situations, and descriptions of artifacts and artifact fragments can be improved significantly. Furthermore, our process adds value to the both the configuration and the composition approach since it particularizes the individual concepts used for specifying situational characteristics (cf. table 1).
- *Method, situation-specific.* This type of method is applicable only in one or more situations for which it was specifically developed. As one can see in the context vs. project type matrix (cf. figure 4), the characteristic trait of situation-specificity can either consist in context-specificity (method is specific to one particular context but applicable for multiple project types, cf. e.g. situations 2 and 6

in the illustration), project type-specificity (method is specific to one particular project type but applicable for multiple contexts, cf. e.g. situation 1 in the illustration), or simultaneous context-specificity and project type-specificity (method is specific to one situation consisting of one context and one project type, cf. e.g. situations 3, 4, 5, 7, and 8 in the illustration). Situation-specific methods represent the output of any approach to situational ME, be it situational method configuration or situational method composition. The differentiation between context and project type can help not only in the process of engineering methods but also in describing the scope of method (or method fragment) applicability [5, 8].

- *Method, valid in all situations.* This type of method is applicable in all situations (i.e. in all combinations of context and project type) that have been identified in step 3. Consequently, this type of method represents a generalization of situation-specific methods as outlined before.

Regardless of the method type, the method has to be validated by applying it in a real world situation for which it is claimed to be valid.

Example. Since the most probable situation for method application is a (more or less) mature DWH competence center that is responsible for costs only, a method for implementing an IT costing process has to be developed and validated for this situation only.

6 Conclusion and Further Research

Our discussion of existing ME approaches showed that nearly all approaches claim to incorporate situational factors. Nevertheless, the existing engineering methodologies do not detail what these situational factors comprise of and how they can be identified. In our paper, we have presented a novel approach to situational ME. Based on the conclusion that context and project type are different determinants of method applicability, we have proposed a procedure model that incorporates both context and project type into situational method construction processes. Our procedure model guides the identification of relevant context and project type factors, examines their occurrences in practice, and classifies them into situations defined as tuple of context and project type. For these situations, suitable methods can be constructed that might be more appropriate than generic methods focusing solely on single situational aspects that are a subset of our definition at the most. With evidence about the frequency of occurrence of contexts and project types in practice, the method engineering discipline can concentrate on developing methods for the most common (i.e. most relevant) situations.

Based on the initial work presented in this paper as well as in our previous work (cf. [15]), three broad categories of research opportunities exist:

- First, our process needs to be validated at large. In the paper at hand, we have proposed a procedure and provided substantial reasons for its meaningfulness based on an extensive literature review as well as on our own experience. In order to prove the practicability and feasibility of the process, we have evaluated single steps (in particular steps 2 and 3) for different domains (cf. e.g. [48, 53]). The validation of the process as a whole is subject to further research.

- Secondly, our process could be extended to cover epistemologically valid method construction processes by evaluating and combing existing ME approaches described in section 2. This opportunity for further research has been sketched in the description of step 4 of our process. However, we believe that further work is necessary. The development of a reference process or even of a method for method construction itself (a so-called "meta method") would help method engineers to design new methods more systematically.
- Thirdly, a reference information model of method fragments and their context(s)/project type(s) could be designed. As stated in section 5, the differentiation between context and project type can be of value for describing the scope of method (or method fragment) applicability. The existence of a reference information model would enable storing and managing situational method elements in an electronic method base, and therefore enhance the potential for reusability of method fragments.

7 References

1. H.A. Simon, *The Sciences of the Artificial* (MIT Press, Cambridge, 1969).
2. A.R. Hevner, S.T. March, J. Park, and S. Ram, Design Science in Information Systems Research, *MIS Quarterly*, **28**(1), pp. 75-105 (2004).
3. S.T. March and G.F. Smith, Design and Natural Science Research on Information Technology, *Decision Support Systems*, **15**(4), pp. 251-266 (1995).
4. J.G. Walls, G.R. Widmeyer, and O.A. El Sawy, Building an Information System Design Theory for Vigilant EIS, *Information Systems Research*, **3**(1), pp. 36-59 (1992).
5. S. Brinkkemper, Method Engineering - Engineering of Information Systems Development Methods and Tools, *Information and Software Technology*, **38**, pp. 275-280 (1996).
6. S. Alter, Work Systems and IT Artifacts - Does the Definition Matter?, *Communications of the Association for Information Systems*, **17**(14), pp. 299-313 (2006).
7. S. Alter, 18 Reasons Why IT-reliant Work Systems Should Replace "The IT Artifact" as the Core Subject Matter of the IS Field, *Communications of the Association for Information Systems*, **12**(23), pp. 366-395 (2003).
8. F. Harmsen, *Situational Method Engineering* (Moret Ernst & Young Management Consultants, Utrecht, 1997).
9. K. Kumar and R.J. Welke, Methodology Engineering - A Proposal for Situation-specific Methodology Construction, in: Challenges and Strategies for Research in Systems Development, edited by W. Cotterman and J.A. Senn (John Wiley & Sons, 1992), pp. 257-269.
10. K. van Slooten and B. Hodes, Characterizing IS Development Projects, in: Method Engineering - Principles of Method Construction and Tool Support, edited by S. Brinkkemper, K. Lytinnen, and R.J. Welke (Chapman & Hall, 1996), pp. 29-44.
11. F.E. Fiedler, A Contingency Model of Leadership Effectiveness, *Advances in Experimental Social Psychology*, **1**, pp. 149-190 (1964).
12. B. Arinze, A Contingency Model of DSS Development Methodology, *Journal of Management Information Systems*, **8**(1), pp. 149-166 (1991).
13. R.J. Schonberger, MIS Design - A Contingency Approach, *MIS Quarterly*, **4**(1), pp. 13-20 (1980).

14. P. Weill and M.H. Olson, An Assessment of the Contingency Theory of Management Information Systems, *Journal of Management Information Systems*, **6**(1), pp. 59-85 (1989).
15. T. Bucher and M. Klesse, Contextual Method Engineering, University of St. Gallen, Institute of Information Management, Working Paper, 2006.
16. M. Heym, Methoden-Engineering - Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme, University of St. Gallen, Ph.D. Thesis, 1993.
17. T. Gutzwiller, *Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen* (Physica, Heidelberg, 1994).
18. T. Bucher, S. Kurpjuweit, and B. Dinter, Risikomanagement im Data Warehousing - Situative Komposition einer methodischen Vorgehensweise, in: DW2006 - Integration, Informationslogistik und Architektur, edited by J. Schelp, R. Winter, U. Frank, B. Rieger, and K. Turowski (Gesellschaft für Informatik, Bonn, 2006), pp. 35-59.
19. F. Karlsson and P.J. Ågerfalk, Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets, *Information and Software Technology*, **46**(9), pp. 619-633 (2004).
20. F. Karlsson, P.J. Ågerfalk, and A. Hjalmarsson, Method Configuration with Development Tracks and Generic Project Types, Paper Accepted for the 6th CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in System Analysis and Design (EMMSAD'01), http://citeseer.ist.psu.edu/503218.html, 2001.
21. U. Baumoel, Strategic Agility through Situational Method Construction, in: Proceedings of the European Academy of Management Annual Conference (EURAM2005), edited by R. Reichwald and A.S. Huff (http://www.euram2005.de, 2005).
22. S. Brinkkemper, M. Saeki, and F. Harmsen, Assembly Techniques for Method Engineering, in: Proceedings of the 10th International Conference on Advanced Information Systems Engineering (CAiSE'98), Springer, LNCS 1413, 1998, pp. 381-400.
23. T. Punter and K. Lemmen, The MEMA-Model - Towards a New Approach for Method Engineering, *Information and Software Technology*, **38**(4), pp. 295-305 (1996).
24. J. Ralyté, Ingénierie des Méthodes a base de Composants, Université Paris 1 - Sorbonne, Ph.D. Thesis, 2001.
25. C. Rolland and N. Prakash, A Proposal for Context-Specific Method Engineering, in: Method Engineering - Principles of Method Construction and Tool Support, edited by S. Brinkkemper, K. Lytinnen, and R.J. Welke (Chapman & Hall, 1996), pp. 191-207.
26. L.J.B. Essink, A Modelling Approach to Information System Development, in: Information Systems Design Methodologies - Improving the Practice, edited by T.W. Olle, H.G. Sol, and A.A.V. Stuart (North-Holland, 1986).
27. L.J.B. Essink, A Conceptual Framework for Information Systems Development Methodologies, in: First European Conference on Information Technology for Organizational Systems (EurInfo 1988), edited by H.J. Bullinger (North-Holland, 1988).
28. M. Franckson, The Euromethod Deliverable Model and Its Contribution to the Objectves of Euromethod, in: IFIP-TC8 International Conference on Methods and Tools for the Information Systems Life Cycle, edited by A.A.V. Stuart and T.W. Olle (North-Holland, 1994), pp. 131-149.
29. C. Braun, F. Wortmann, M. Hafner, and R. Winter, Method Construction - A Core Approach to Organizational Engineering, in: Proceedings of the 20th Annual ACM Symposium on Applied Computing (SAC 2005), edited by H.L. Haddad, Lorie M.; Omicini, Andrea; Wainwright, Roger L. (ACM, Santa Fe, 2005), pp. 1295-1299.
30. N. Prakash, On Method Statics and Dynamics, *Information Systems Journal*, **24**(8), pp. 613-637 (1999).
31. D. Truex and D. Avison, Method Engineering - Reflections on the Past and Ways Forward, Proceedings of the Ninth Americas Conference on Information Systems (AMCIS 2003), 2003.

32. J. Ruegg-Stuerm, *Das neue St. Galler Management-Modell - Grundkategorien einer integrierten Managementlehre* (Paul Haupt, Bern, 2002).
33. W.H. Inmon, *Building the Data Warehouse* (Wiley, New York, 2002).
34. W.H. Inmon, J.D. Welch, and K.L. Glassey, *Managing the Data Warehouse* (Wiley, New York, 1997).
35. R. Kachur, *Data Warehouse Management Handbook* (Prentice Hall, 2000).
36. B. Ives, S. Hamilton, and G.B. Davis, A Framework for Research in Computer-Based Management Information Systems, *Management Science*, **26**(9), pp. 910-934 (1980).
37. R. Winter and M. Meyer, Organization of Data Warehousing in Large Service Companies - A Matrix Approach Based on Data Ownership and Competence Centers, Proceedings of the Seventh Americas Conference on Information Systems (AMCIS 2001), 2001.
38. C. Limacher, Organisationskonzept und Erfolgsfaktoren fuer den Betrieb und die Weiterentwicklung einer Data Warehouse Loesung am Beispiel des Universitaetsspitals Zuerich, University of St. Gallen, Bachelor Thesis, 2005.
39. M. Cossentino, S. Gaglio, B. Henderson-Sellers, and V. Seidita, A Metamodelling-based Approach for Method Fragment Comparison, in: The 18th International Conference on Advanced Information Systems Engineering, edited by T. Latour and M. Petit (Presses Universitaire de Namur, Luxembourg, 2006), pp. 419-432.
40. Office of the Government of Commerce, *Introdction to ITIL* (Stationary Office, London, 2005).
41. J. Cao, J.M. Crews, M. Lin, A. Deokar, J.K. Burgoon, and J.F. Nunamaker Jr, Interactions between System Evaluation and Theory Testing - A Demonstration of the Power of a Multifaceted Approach to Information Systems Research, *Journal of Management Information Systems*, **22**(4), pp. 207-235 (2006).
42. E.J. Webb, D.T. Campbell, R.D. Schwartz, L. Sechrest, and J.B. Grove, *Unobtrusive Measures - Nonreactive Research in the Social Sciences* (Rand McNally, Chicago, 1966).
43. R. Lamb and R. Kling, Reconceptualizing Users as Social Actors in Information Systems Research, *MIS Quarterly*, **27**(2), pp. 197-235 (2003).
44. R.B. Kline, *Principles and Practice of Structural Equation Modeling* (Guilford Press, New York, 2005).
45. C.A. van Lengen and J.N. Morgan, Chargeback and Maturity of IS Use, *Information & Management*, **25**(3), pp. 155-163 (1993).
46. W.P. McKinnon and E.A. Kallman, Mapping Chargeback Systems to Organizational Environments, *MIS Quarterly*, **11**(1), pp. 5-20 (1987).
47. D.M. Ahern, A. Clouse, and R. Turner, *CMMI Distilled - A Practical Introduction to Integrated Process Improvement* (Addison Wesley, 2003).
48. M. Klesse and R. Winter, Organizational Forms of Data Warehousing - An Explorative Analysis, in: Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS-40), edited by IEEE Computer Society (IEEE Computer Society, Los Alamitos, 2007).
49. M.J. Earl, *Management Strategies for Information Technologies* (Prentice Hall, 1989).
50. B.G. Tabachnick and L.S. Fidell, *Using Multivariate Statistics* (Allyn & Bacon, 2006).
51. P. Chamoni and P. Gluchowski, Integrationstrends bei Business-Intelligence-Systemen - Empirische Untersuchung auf Basis des Business Intelligence Maturity Model, *Wirtschaftsinformatik*, **46**(2), pp. 119-128 (2004).
52. M. Klesse, Leistungsverrechnung im Data Warehousing - Ergebnisse einer empirischen Studie, University of St. Gallen, Institute of Information Management, Working Paper, 2005.
53. T. Bucher and R. Winter, Classification of Business Process Management Approaches - An Exploratory Analysis, *BIT - Banking and Information Technology*, **7**(3), pp. 9-20 (2006).