Situation vs. Context: Considerations on the Level of Detail in Modelling Method Adaptation

Christian Janiesch
SAP Research CEC Brisbane
SAP Australia Pty Ltd, South Brisbane, Queensland, Australia
Email: c.janiesch@sap.com

Abstract
Creating a universal conceptual modelling method, which can be used without modification in all situations, is not feasible. Appropriate methods for problem solving must be chosen, adapted or designed depending on the characteristics of the (pre-)supposed application. However, analyses reveal that the situativeness of documented adaptations is not as situational as method engineering theory tends to put it. This paper acts as a primer and provides a survey of the state-of-the-art to systematise the diverse field of (modelling) method engineering. We focus on the cause of adaptation and suggest regarding the overall context of method deployment, rather than focusing on situational details of its application. We conclude by proposing cornerstones of a research agenda to formalise and standardise the definition of context and, thus, make it more manageable.

1. Context and situation in conceptual modelling

Information systems (IS) support the coordination of all information and communication flows in any given domain. This is done under the crucial condition of meeting requirements for economical efficiency. In order to facilitate the information systems development (ISD) process any development group must agree on some shared representation forms. Based on this representation, ideas, thoughts, opinions, objectives, and beliefs about the object system can be exchanged and discussed [27]. One suitable means of representation is considered to be conceptual modelling.

Socio-technical approaches to design science research emphasise that IS are man-machine systems, which comprise behavioural sub-systems, for instance people or culture, as well as technological sub-systems, for instance hardware or software [10]. Information technology (IT) artefacts as the output of design science research, such as models or methods, are intended to solve organisational problems [40]. This entails that any situated use thereof can only function properly when embedded in its socio-technical context.

Hence, it has to be accepted that a universal method, which can be used everywhere without modification, is not feasible [12, 34, 57]. Instead, appropriate methods for problem solving must be chosen, adapted or designed depending on the specific characteristics of their application. In the IS discipline, the research areas of domain specific method engineering [39] and situational method engineering (SME) [12, 26, 35] deal with this idea of method tailoring.

It is understood that the requirements for this adaptation arise from the situation, i.e. the combination of circumstances at a given moment [26]. The assumption of this research is, however, that the use of the situation might be too detailed to allow for the meaningful reuse and comparison of resulting models. We propose to regard the more general context of method deployment, rather than focusing on situational details. In this way, a major part of the method can be contextualised beforehand based on recurring domain- and purpose-specifics while singular and situational ad hoc adaptation may take place subsequently on an individual level.

This paper is structured as follows: first, we describe the underlying understanding of conceptual modelling as well as method engineering in the context of ISD in detail. Following up from research concerned with SME, the idea of conceptual modelling methods based on a context-based approach is introduced and justified. We then underpin our research assumptions with a quantitative analysis of journal articles. We conclude by providing cornerstones of a research agenda to formalise and standardise this diverse field of research to provide a better and less arbitrary way for specifying context and method rationale.
2. Understanding the context of this work

2.1. Conceptual models in Information Systems Development

Conceptual models feature three main characteristics. These are mappings of representations, abstractions from originals, and in compliance with pragmatic requirements. Thus, in the modelling process multiple people’s perceptions of a matter are typically condensed into a shared representation. Thereby, the models drawn upon always result in an abstract account of reality.

Conceptual models are usually graphical, i.e. semi-formal, representations [cf. also 36] and can be applied to static (e.g., data models) and dynamic (e.g., process models) states of affairs in some domain [64]. Generally they are used to structure and systematise problems and thereby used to omit irrelevant aspects of the surrounding scenario and help focus on the key problem at hand. Thus, a conceptual model is the representation of an application domain for the ends of a subject which is commonly based on a semi-formal language with a graphical representation [65].

Following Wand and Weber [64], conceptual modelling serves in particular to support communication between developers and users, to help analysts understand a domain, to provide input for the design process, and to document the original requirements for future reference. Usage in the early stages of ISD is considered to be particularly beneficial, since the efforts for resolving mistakes made in this stage increase exponentially as time passes and subsequent project stages commence [cf. 42].

Evidently these different purposes require different modelling processes. If, for instance, conceptual models are intended to serve as an input for the design process, a formal and unambiguous grammar has to be used in order to map concepts precisely to implementation artefacts. Support for the communication process, however, can be achieved by less formal means, e.g. in tabular form [for other representation forms cf. e.g. 58].

2.2. Methods in Information Systems Development

Problems can be characterised as the discrepancy between an as-is and a to-be situation. Methods describe systematic procedures to overcome problems [7]. They provide a systematic way of acting, which leads to reusable results, and are built on a set of rules that provide solutions to a specific problem type. Constituent elements of methods are actions, artefacts, constructs, notations, and roles [11, 31]. A method should be tool supported [11, 12, 26].

Roles carry out actions. Actions are related to each other in the form of a network and can be nested to represent processes or, on a higher level of abstraction, procedure models. Within an action, artefacts, such as class diagrams or activity diagrams, are produced and consulted to retain or understand the involved information. Thus, an artefact is the output of an action and can also be the input for a subsequent action. Artefacts are described by notations. Notations represent constructs, which provide the formal vocabulary to describe a certain domain. Also, constructs can be nested to form conceptual modelling methods (i.e. their meta model). A set of actions, also termed technique, describes how to use it. Conceptual modelling methods are commonly perceived as key elements of methods.

There is a broad range of different conceptual modelling methods suitable for different purposes and specific domains [63]. However, it is widely acknowledged that there is a vast variety of competing and complementary conceptual modelling methods [43].

The construction of conceptual modelling methods is commonly regarded as an engineering task. Consequently, in the IS discipline, the term method engineering has been established to encompass all activities concerned with the creation of methods in general and conceptual modelling methods in particular. The idea was introduced by Bergstra et al. [9], refined by Kumar and Welke [35], and for the course of this discussion usefully defined by Brinkkemper [12] as “the engineering discipline to design, construct, and adapt methods, techniques, and tools for the development of information systems.”

Method engineering is commonly understood as a three phased process (also lifecycle) consisting of method requirements engineering, method design, and method construction, implementation, and evaluation [22, 24]. Since, by definition, method engineering also encompasses method adaptations, another more software engineering-oriented understanding of method engineering process is the following: method selection, method construction, and tool adaptation [12, 26].

As mentioned initially, it is widely accepted that a universal method, which can be used without modification in all situations, is not feasible. Rather, appropriate methods for problem solving must be chosen, adapted, or designed depending on the specific characteristics of a situation. The definition
of Brinkkemper does not explicitly exclude the tailoring of methods to suit situational circumstances. It is, however, prevailing opinion that method engineering is a process that does not intend to produce situational solutions to software development problems. Rather, the term situational method engineering [12, 26, 35] has been used to voice this special circumstance in the method engineering community.

### 2.3. Situational method engineering in Information Systems Development

Starting in the 1970s, many new conceptual modelling methods were developed to improve the efficiency of ISD processes [43]. Unfortunately, the engineering process for conceptual modelling methods was strongly based on individual intuition and lacked methodological and theoretical foundation [64]. This proliferation continued into the 1980s until the research community became dissatisfied with the situation and the permanent evaluation of new methods. The term yet another modelling approach (YAMA) was coined [46].

The emergence of the situation can be traced to the dissatisfaction of researchers and practitioners with method efficacy which led to an increased use of informal and proprietary methods [19, 44]. This local method development is characterised by a high degree of flexibility and situational fit but fails to exhibit the ability to provide useful assistance for more than one situation [58].

Consequently, the design process for software development methods in general and conceptual modelling methods in particular was scrutinised. Situational method engineering emerged with the aim to provide guidance on the modification of methods to suit diverse requirements [12, 35] retaining the ability to be a manageable multi-purpose method. As Tolvanen [58] states, “methods have to be maintained and revised.”

The approaches of this stream of research may seem rather diverse, but Becker et al. [7] showed that they can be systematised according to the method’s approach for knowledge reuse. These approaches comprise patterns [1], components [24], modules [30], views [66], and reference models [6]. All approaches can be found in a similar form not only in conceptual modelling or method engineering but also in software engineering.

Each of these approaches uses different mechanisms to adapt the original artefact to a situational artefact. The mechanisms can be divided into analogy construction, aggregation, configuration, specialisation, instantiation [6, 62], and restriction [22]. For an allocation of mechanism to approaches cf. Table 1.

At present, time tends to repeat itself and the creation of yet another method adaptation approach (YAMA2) can be observed [e.g., 4, 5, 15, 24, 26, 31, 48]. It is reasonable to assume that some sort of consolidation will take place.

### 3. Actual use of context in IS

#### 3.1. Method contextualisation

The foundation and justification to include context in conceptual modelling methods can be seen in the socio-technical design approach to ISD. It accounts for the subjective nature of perception and accordingly for the subjective construction of representation forms [41]. Conceptual modelling is a process that is usually conducted by system analysts in collaboration with domain experts. Both materialise their perception of the real world into an abstract representation. The model is hallmarked by the mindsets and intentions of the participating people and therefore cannot claim to represent some objective reality. Hence, conceptual modelling is a social process [20] and assists the discourse of actors. Therefore, it needs to be treated as such.

---

### Table 1. Mapping of reuse mechanisms to reuse approaches (cf. [7])

<table>
<thead>
<tr>
<th>Component (CO)</th>
<th>Module (M)</th>
<th>Pattern (P)</th>
<th>Reference Model (RM)</th>
<th>View (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analogy Construction (AC)</td>
<td>---</td>
<td>P/AC</td>
<td>RM/AC</td>
<td>---</td>
</tr>
<tr>
<td>Aggregation (A)</td>
<td>CO/A</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Configuration (C)</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Instantiation (I)</td>
<td>---</td>
<td>M/I</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Specialization (S)</td>
<td>---</td>
<td>P/S</td>
<td>RM/S</td>
<td>---</td>
</tr>
<tr>
<td>Restriction (R)</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>V/R</td>
</tr>
</tbody>
</table>

---
The interdisciplinary approach of critical discourse analysis proposes a three-dimensional framework to discourse analysis: text, text production and consumption, and events of socio-cultural practice [17], i.e. transferred to the problem domain of conceptual modelling: model, method, and their context as the pragmatic representation of social practice. While procedure models, modelling conventions, and ontologies as means to capture domain knowledge exist to capture model semantics [28], context is at the heart of pragmatics. Therefore, since “meaning is at the heart of both semantics and pragmatics” [18], it is important not to neglect context.

Consequently, when regarding modelling methods as social and contextualised complexes, it becomes necessary to include some stance of context in the meta model. In linguistics, the notion of reference spaces [23] or context spaces [51] is used to describe context within script. Similarly, models or parts thereof can be equipped with context.

### 3.2. Understandings and applications of context in IS

As outlined earlier, we believe that the very singular requirements of a situation might be too granular to allow for resulting conceptual modelling methods which can be compared and reused. Therefore, we propose to pre-configure methods based on their broader application context. However, in IS the understanding of context is often blurred as Section 2 revealed. In this section we attempt to systematise the current perceptions of context as relevant parameters for the adaptation of conceptual modelling methods. We begin by presenting a broad overview of the understanding of context and continue with a focus on IS.

Context is hard to grasp or as Asher [2] phrases it: “Context is one of those linguistic terms which is constantly used in all kinds of context but never explained.” A context is a complex of circumstances, conditions, and influence factors. In the case of models, context describes or represents the environment in which the model or a component thereof has a certain meaning. It is a theoretical construct to interpret meaningful expressions of discourse [61]. However, “due to the large variety of research fields, it is no surprise that there is no general agreement on a unique, shared notion of a context. Rather the interpretation of the notion itself is context-sensitive” [37].

In the IS discipline several notions of context have surfaced. Each of them includes several distinctions which further detail the composition of a context. While Bucher et al. [14] relate context and project type in a matrix to determine situational characteristics, Leppänen [38] and Becker et al. [8] introduce individual factors that have an impact on the performance of methods in different environments. Leppänen [38] uses seven so-called domains to differentiate influence on context in his four-dimensional framework of perspectives, domains, and workflows on method engineering and ISD respectively. Becker et al. [8] propose a two-staged method engineering process which includes adaptation to more general requirements (i.e. purpose and domain of the method) as well as situational details (i.e. company and project). Rosemann et al. [52] do not explicitly differentiate individual factors. They regard them implicitly by defining four types of context which encompass each other. This entails that influences are differentiated by their proximity to the artefact where each of the outer context types may have effects on the inner context types. Aydin [3] understands context as something that constitutes a situation based on the intentions of a human actor called agency. However, most authors grasp context as something more general, i.e. “the set of circumstances or facts that surround a particular event, situation, etc.” [50]. Dey et al. [16] apply context to the domain of computing devices and applications. Due to the broad diversity of context information, in their opinion it is useful to categorise context for a better, systematic comprehension of context. For this purpose, they define the four basic context categories as identity, location, status, and time. Saidani and Nurcan [54] propose an approach to context-aware business process modelling which is a relatively new research area that extends upon what was commonly named as configurable reference modelling [53]. They use basically the same categories. Similar approaches to context-aware business process management focus on process execution [25] and process analysis [47]. UN/CEFACT proposes a context model consisting of eight context driver categories that are populated by tree structure lists such as country codes etc. The Unified Context Methodology project extends upon this model [60]. While this is an important initiative, the outcome will most likely produce a very comprehensive model that will ease the definition of what we call situation but may be too granular for defining a context.

Table 2 summarises this exemplary overview of context interpretations, for an overview of current interpretations of situation cf. also Bucher et al. [14].
3.3. Theory and practise of situation and context is IS literature

However important the issue of method adaptation to a specific situation may seem for practice, an intense review of renowned international journals reveals that although the problem is well established in the community, the documentation of individual cases is rather poor. For this analysis, an equal amount of articles from U.S. and European journals has been reviewed. The following journals have been considered for their documentation of ISD cases: MIS Quarterly (MISQ), Information Systems Research (ISR), Information Systems Journal (ISJ), European Journal of Information Systems (EJIS), and Business & Information Systems Engineering (BISE)\(^1\). The choice of journals was motivated by their impact factor on the IS discipline as well as their focus of research. We did not focus on method engineering or modelling journals in particular in order to get a more realistic understanding of the actual perception and practice of method adaptation outside the core community. Data Collection started with issue 1/1997 and ended with the last issue of 2006. In Table 3 the initial findings are aggregated. Column 3 lists the number of articles reviewed per journal (percentage is based on all reviewed articles). Column 4 lists the articles thereof which are based on practical case studies (from here, the percentage is always according to total articles per journal). Column 5 further details the cases into those where

\(^1\) Formerly known as Wirtschaftsinformatik (WI).
some modelling took place. Column 6 finally includes only those articles where SME was explicitly exercised.

Since method engineering is a European rather than an American tradition, the amount of explicit modelling and method engineering cases was expected to be lower in the corresponding U.S.-based journals. However, as Table 3 shows, not only are the U.S. journals rather close-mouthed about method engineering, but only a few documented cases of a controlled adaptation of methods can be found in European journals, although the number of practical cases is rather high. In addition to that, some of the method engineering cases included in this table offer only scenarios or theoretical essays on the subject which introduce only minor exemplary applications.

If adaptations have taken place though, they targeted the notation of the conceptual modelling method in roughly 51% of the cases; the procedure model to use the modelling method was adapted in about 28% as was the adaptation of the meta model with 25%.

Harmsen [26] introduced the concept of controlled flexibility as a basis for SME to circumvent the proliferation of countless variants. Controlled flexibility accommodates change with standardised and proven building blocks on the basis of a uniform terminology in order to maintain a stable and manageable repository of the core method. In particular, the notion of standards is stressed to provide unambiguous and recognisable components to provide higher acceptability of the final method.

Harmsen’s, as well as most other approaches to SME aim at producing a situational method at a very granular level to fit very specific and, therefore, mostly singular situations [14].

Thwarting this idea, however, the above analysis of journal articles also reveals that the situativeness of the documented adaptations was not as situational as method engineering theory tends to put it. We used the adaptation model of Becker et al. [8] to categorise the change. In brackets we give examples to illustrate the categories.

Only about 10% of the adaptations were due to either the specific project or the targeted application system (e.g. Oracle or SAP ERP system). Equally marginal, in only 6% of the cases, the method was explicitly adapted to a specific company (e.g. its size, culture, etc.). The majority of the adaptations were due to the purpose (29%) (e.g., workflow management as opposed to process reengineering for process modelling) or the domain (44%) of application (e.g. retail, automotive, health care, e-government, etc.).

These numbers propose that while situational method adaptation is still necessary, a controlled pre-adaptation of methods according to factors such as purpose or domain of application may be beneficial and ease the overall adaptation effort and the repeatability of the adaptation task. This observation goes along with the Pareto principle observed in other areas and would promise more economical (situational) method adaptations if they are based on context-specific method variants.

<table>
<thead>
<tr>
<th>Reuse Mechanism</th>
<th>% of SME Approaches</th>
<th>Reuse Approach</th>
<th>% of Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation</td>
<td>70 %</td>
<td>Components</td>
<td>85.7 %</td>
</tr>
<tr>
<td>Instantiation</td>
<td>5 %</td>
<td>Modules</td>
<td>0 %</td>
</tr>
<tr>
<td>Analogy Construction</td>
<td>15 %</td>
<td>Patterns</td>
<td>3.6 %</td>
</tr>
<tr>
<td>Specialization</td>
<td>45 %</td>
<td>Reference Models</td>
<td>10.7 %</td>
</tr>
<tr>
<td>Configuration</td>
<td>15 %</td>
<td>Views*</td>
<td>3.6 %</td>
</tr>
</tbody>
</table>

* The mechanism of restriction was not included in [7].
There is also a certain imbalance in the utilization of these approaches in method engineering (cf. Table 4). The left part of the table originates from an analysis of 20 method engineering approaches by Becker et al. [7], and the right part links data from the above review of journal articles. Both analyses support each other in terms of scale as the linking lines show.

Consistent with this observation, in SME the prevailing approach is not to consider a method as a single monolithic block. Instead, methods consist of a set of fragments [e.g., 13, 26], also called chunks [e.g., 49] or components [e.g., 33]. These blocks can have different levels of granularity and can describe the product (which is created) as well as the process (how is it created) of a method. The blocks can comprise a single activity or construct but they can also contain a complete method. Hence, a situational method engineering project can start with a set of atomic method blocks, which must be assembled, as well as an existing method which has to be modified [48]. However, so far method engineering research has mainly focused on assembling methods. It is only recently, that other approaches have manifested.

4. Contextualisation of conceptual modelling methods: A research agenda

In order to provide an efficient means of conceptual modelling several cornerstones have to be in place. Based on evidence from the literature review and the theoretical grounding, it can be concluded that incorporating some concept of context in conceptual modelling methods is beneficial for the understanding of their content. While this has been generally acknowledged, there is no agreement on the actual composition of context. It is however widely agreed upon that the use of standard modelling notations such as UML and BPMN is more beneficial for a project than an arbitrary use of notation and grammar to model [29]. Also, the use of standardised method adaptation mechanisms improves the traceability and consistency of models. Independent of the SME approach, the mechanisms were formalised and recommendations can be inferred for their suitability depending on the required degree of guidance and degree of preparation [7].

We extend upon this by saying that not only the source material of the conceptual modelling method and the mechanisms to adapt it should be well understood and formalised, but also the rationale of the adaptation. By rationale, we mean the situational or contextual circumstances that necessitate the adaptation of the method. As the analysis in Section 3 revealed, there is no uniform understanding of the actual specification of context let alone general categories for structuring. Neither has an attempt to define the necessary level of detail been made. In most approaches, the entities that constitute context are hardly formalised, exhaustive, or properly derived. They are usually provided as anecdotal evidence to support the case at hand.

In order to provide a suitable context model, a number of design issues have to be resolved:

(a) Define a structural model for context information
(b) Define appropriate context instance information for these categories
(c) Define the necessary level of abstraction for the adaptation
(d) Define a procedure model to support the context definition process

As Section 3 showed, a number of proprietary categorisation models for contextual/ situational information exist. To tackle issue (a), we propose to combine a layered approach such as [52] and [8] with the category-based approaches such as [38] and [59]. While categorisation allows systematisation of context factors, the layering allows for better support of different levels of detail in (pre-)adaptation. Based on the examined models, we propose to have at least an external/ environmental context that covers topics such as purpose and domain/ industry as well as an internal/ immediate situation that covers project/ company/ application systems specifics. Similar approaches can be observed e.g. for process modelling [45] and business intelligence [55]. What they have in common is that they focus on multiple levels of one particular purpose and are not intended to provide a general adaptation methodology.

Within these two layers, context categories need to be developed. While there is an attempt to create a comprehensive context classification scheme with instance information (b) within the Unified Context Methodology project, its applicability needs to be revisited due to the difference in focus. Also, no comparison has been done to date with other existing approaches such as [21] and [56].

Be this as it may, the most pressing question is probably centred on issue (c): “Where to draw the line between context and situation?” However, the answer to this question with reference to Leppänen’s comment on context can only be: “It depends.” Thus, the attempt has to be to provide as much information and guidance as possible to make informed decisions.

Setting out from the proposed solution for issue (a), we believe that a contextualised method should be adapted towards external factors only, which are applicable to a number of different use cases. Only
then, if at all, should the conscious decision be made to further adapt it to situational details. A procedural framework for the contextualisation of methods (d) that is aligned with the structure of the context (a) can ease the task and assist in finding the right level of abstraction.

We propose to adapt towards the particular purpose and, in addition, towards the (business) domain of application first. In accordance with the quantitative analysis, the modification should be mainly on the notation or lightweight adaptations of the meta model and procedure model. This would constitute a contextualised method.

For the actual application, it might be necessary to adapt the method ad hoc towards a situational method to serve a certain project goal, or respectively to suit the application system or company where it is to be applied. This would constitute a situationalised method. It is important to keep in mind that the reuse and comparability of the resulting modelling method variants will be impaired due to the very specific adaptations.

As argued above, it is also not feasible to design completely from scratch but to abide by standards to facilitate reuse and model comprehension. Hence, an existing general purpose modelling method should be taken as a basis whenever possible. This is achieved more easily with purpose-spanning and/or domain-spanning methods which are generally available. Modelling patterns or templates can assist the application of the method [32].

Finally, in order to maintain a consistent overview over these method variants, efficacious approaches to method variant management have to be explored. Also, the implementation of the conceptual specification becomes necessary – or as Tolvanen [38] expresses it: “Using meta models without considering their support in ISD tools would be the same as designing an IS without implementing it.”

5. Acknowledgement

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference “01MQ07012”. The authors take the responsibility for the contents.

6. References


