Volume 6, Issue 9, September 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Method Prescription Based on Method Rationale

Rinky Dwivedi
Assistant Professor, Maharaja Surajmal Institute of Technology,
New Delhi, India

Abstract: ***Methods encapsulate rationale (i.e. knowledge about method) and this method rationale forms the basis for particular method prescriptions provided to application engineer by the method engineer so that he can adapt, tailor existing methods or merge various prescribed methods to yield a new methodology that may conform to the needs of project situation at hand. In this paper, it is taken as a requisite that methods are residing in method base that in turn will help in finding right methodology meeting organization demand. Different methods may have different strengths and weaknesses which imposes the need for an analytic tool that will serve the purpose of prescribing appropriate method out of several available at hand.***

Keywords: ***Method prescription, Method rationale, Method configuration, Method engineering, Method fragments, Method chunk, Situational Method Engineering, Method adaptation, Method components, Method tailoring, Meta models, Information systems.***

I. INTRODUCTION

Information systems that are being used for more and more applications are becoming more complicated and expensive, while at the same time affecting large parts of our society. It is therefore important that information systems are developed in an effective and efficient way. To achieve this, numerous methods for information systems development have been described. Such a method typically consists of descriptions of the activities to be performed, the products to be delivered, and the tools to be used.

A problem with these methods is that they hardly take into account the situation in which an information system is developed. Each situation is, in principle, different. A situation affects the way of working and the product types to be delivered. On the one hand, a method should accomplish standardization, on the other hand should it be flexible, to match the situation at hand. Controlled flexibility is achieved by constructing methods.

For each situation, for instance a project or an organization, a different method is built. This method completely takes into account the circumstances (situation factors) applicable in that situation. We call such a method a situational method.

Each method possess some concepts, notions, actions which are collectively termed as method fragments and Method fragments are subdivided into two main categories: conceptual method fragments and technical method fragments. A conceptual method fragments is either a product fragment, or a process fragment. A product fragment describes a product of the information systems engineering process. A process fragment describes the activities of that process. Technical method fragments describe the automated tools (CASE tools) supporting the information systems engineering process, and are subdivided into three categories: tool fragment, repository fragment and process manager fragment.

Besides method fragments, the Method Base contains several other concepts. Actor and actor role are used to model the so-called human aspect of methods, i.e. the persons in methods and their roles and functions. Association is used to model relationships between product fragments. Product fragments play a product role in these relationships. Process fragments play a process role while manipulating product fragments. In addition to this, the Method Base contains the rule concept, to specify constraints on method fragments.

Between method fragments and other concepts a number of relationships and properties is defined, with which method fragments can be characterized and selected. Properties can have a fixed value domain, but can also have a value from a value set that is not stable. From the former type, the so-called stable properties, value domains are given in this dissertation.

To enable the combination of method fragments, heuristics and formalized method assembly rules have been defined. Given a situation and expected success of a project, the heuristics help in choosing suitable method fragments.

The assembly rules are then used to combine method fragments in such a way, that a high quality situational method is made. Besides the suitability requirement, other criteria play a role: completeness, consistency, efficiency, soundness and applicability. We have grouped the method assembly rules by these criteria.

The aim of this search is to provide the method user, a list of methods that are in accordance with the requirements given to the method engineer by the application engineer, on the basis of method rationale.

II. THE CONCEPT OF METHOD RATIONALE

The concept of Method Rationale has previously been defined as "information regarding decisions that leads to a specific Meta model" [1,2,3] This definition originally comes from the field of method engineering and has its foundations in the concept of design rationale[5,6] Basically, design rationale is focused towards possibilities to capture and communicate decisions concerning a design. Notion has been transferred into an ME context with similar meanings. By Meta model [7,8] mean a specific method that can be used in different development situations. The main point they present is that methods in use are subjects to change. The persons using the method make decisions during method use and these decisions shape or adapt the method in an organizational context. Another way to put it would be to say that method use, leads to method adaptation. Rossi et al. discuss two different types of method rationale: method construction rationale and method use rationale. [9,10] focus on method use rationale and how this type of method rationale can be used to explicate otherwise tacit knowledge about method adaptations to co-workers.

III. METHOD ENGINEERING AND NEED FOR METHOD PRESCRIPTION

Method Engineering is the systematic analysis, comparison, and construction of Information Systems Engineering Methods. The term is virtually replaced by situational method engineering now. Situational Method Engineering is the sub-area of Method Engineering directed towards the controlled, formal and computer-assisted construction of situational methods out of method fragments.

To ease the process of creating new method out of existing fragments in order to suit project need, we need method prescription that is completely based on method rationale.

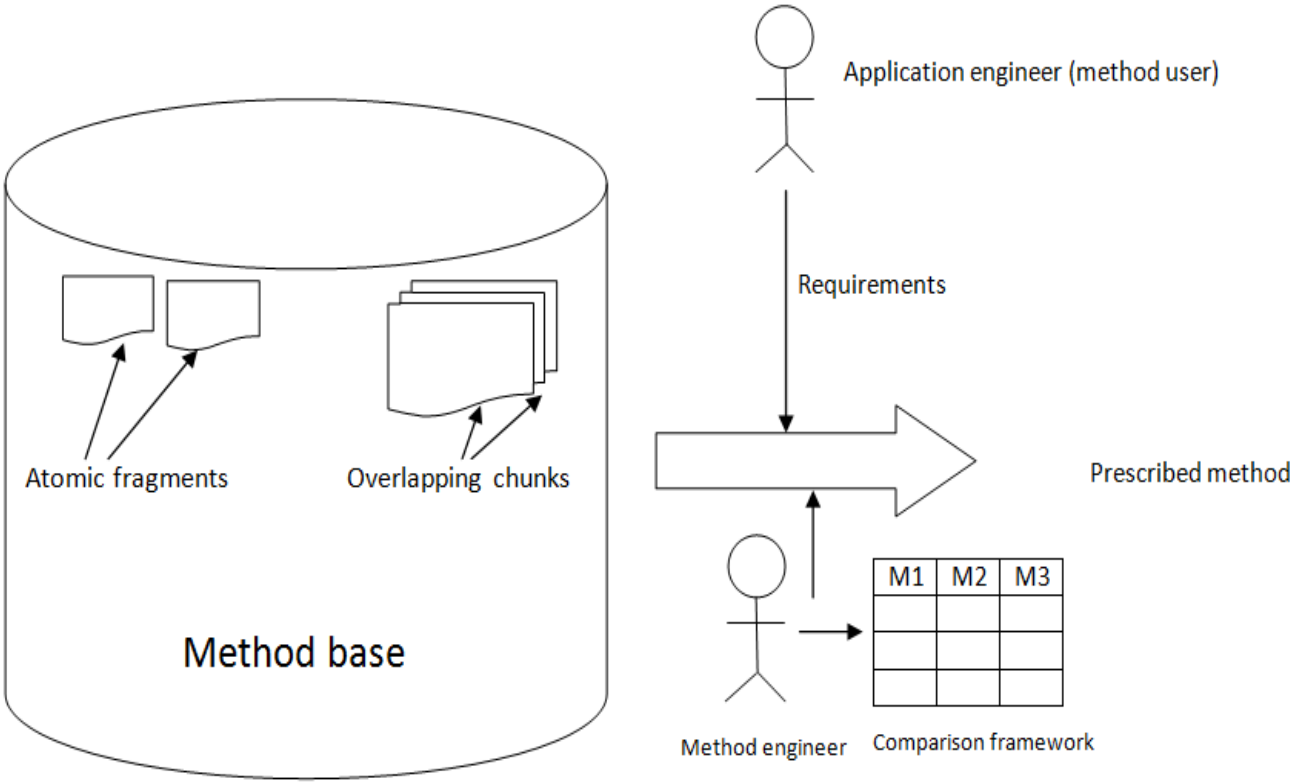


Figure 1: The Process of Method Prescription

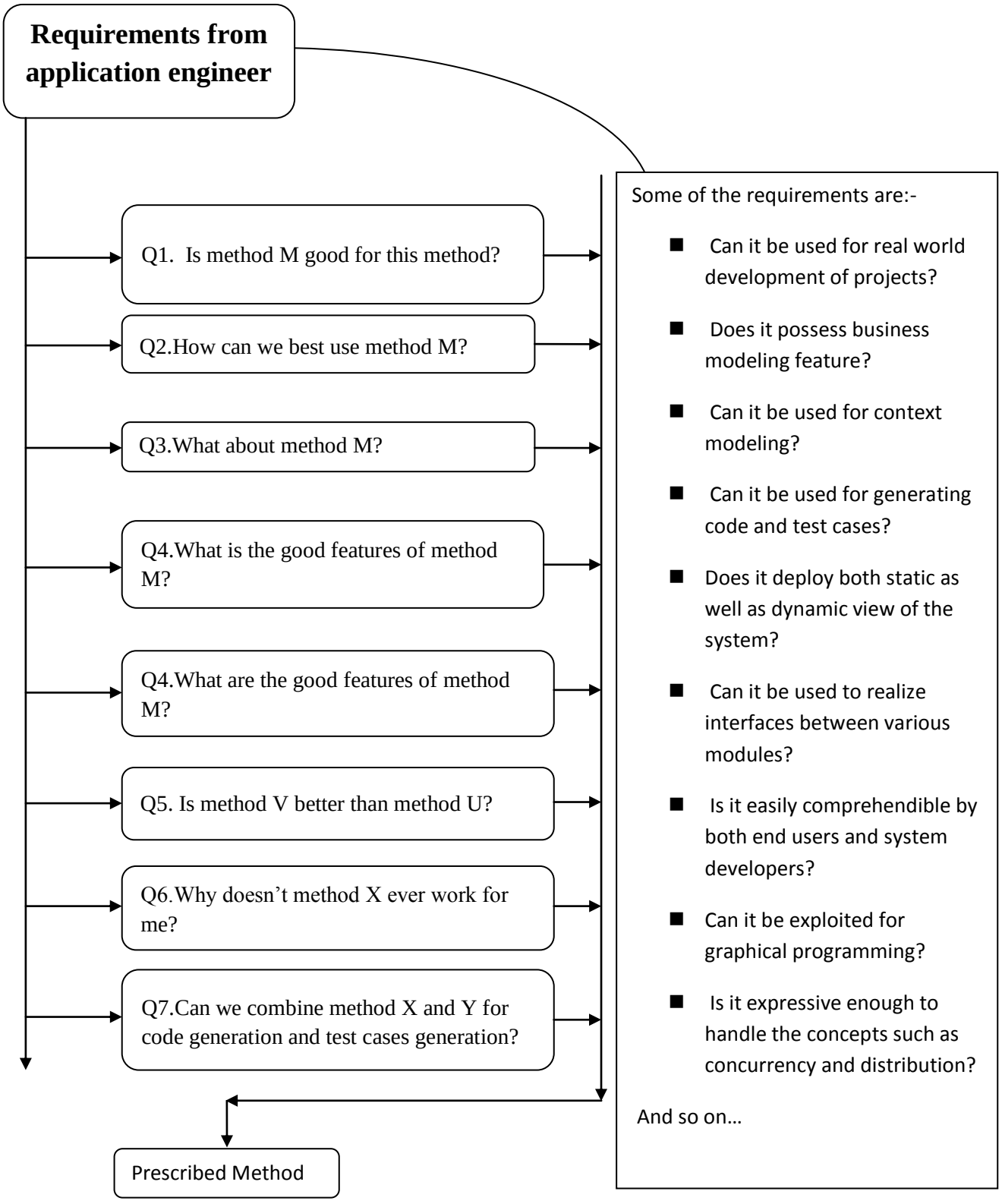
IV. REQUIREMENTS FROM APPLICATION ENGINEER (METHOD USER)

Requirements are gathered by a set of well defined questionnaire. For example:

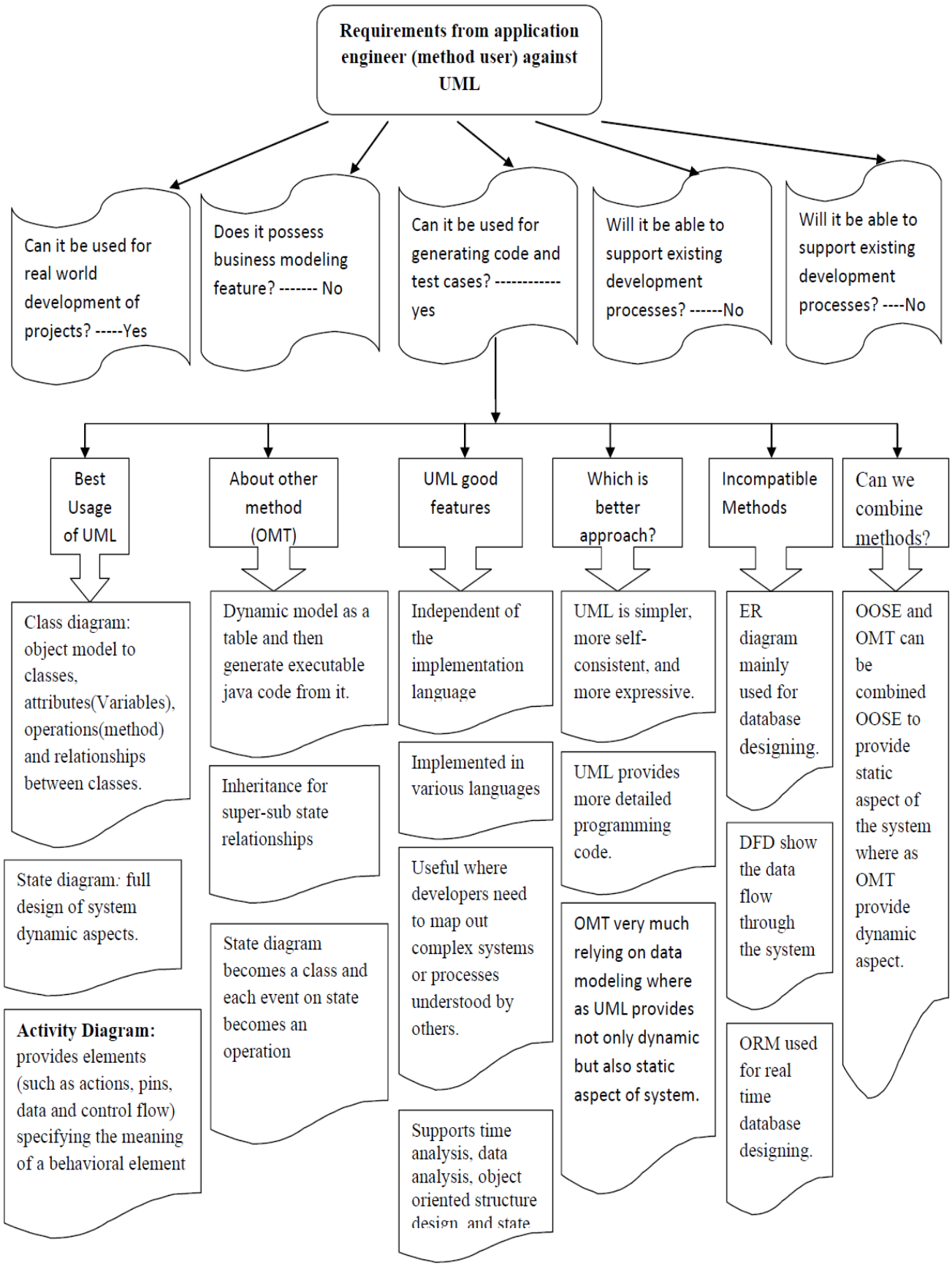
1. Can it be used for real world development of projects?
2. Does it possess business modeling feature?
3. Can it be used for context modeling?

4. Can it be used for generating code and test cases?
5. Does it deploy both static as well as dynamic view of the system?
6. Can it be used to realize interfaces between various modules?
7. Is it easily comprehendible by both end users and system developers?
8. Can it be exploited for graphical programming?
9. Is it expressive enough to handle the concepts such as concurrency and distribution?
10. Will it be able to support existing development processes?

Block diagram



For UML



V. CONCLUSIONS

In this paper we have discussed the concept of method rationale in relation to information systems development for prescribing appropriate methods to the method user. We also provided a comparison between several methodologies based on certain parameters.

A problem with the proposed framework is its rigorous approach. In most practical cases, it is probably impossible, or at least intractable to track all suggested primitives. Consequently, a future task is to develop a process to tailor rationale analysis frameworks at appropriate levels of granularity. However, such 'simplified' frameworks must still be anchored in the framework presented in this paper.

REFERENCES

[1] Harmsen A.F., Situational Method Engineering. Moret Ernst & Young , 1997.

[2] Booch, G., Rumbaugh, J., & Jacobson, I. (1999). The Unified Modeling Language user guide. Harlow, UK:Addison-Wesley.

[3] Brinkkemper, S., Saeki, M., & Harmsen, F. (1999). Metamodelling based assembly techniques for situational method engineering. Information Systems, 24(3), 209–228.

[4] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). Object-oriented modeling and design. Englewood Cliffs: Prentice Hall.

[5] Karlsson, F., Ågerfalk, P. J.: Method Configuration: Adapting to Situational Characteristics While Creating Reusable Assets. Information and Software Technology 46 (2004) 619–633.

[6] Ågerfalk, P. J., Wistrand, K.: Systems Development Method Rationale: A Conceptual Framework for Analysis. In: Proc. 5th International Conference on Enterprise Information Systems (ICEIS 2003) (2003) 185–190.

[7] Fitzgerald, B., Russo, N. L., Stolterman, E.: Information Systems Development:Methods in Action. McGraw-Hill, Berkshire, UK (2002).

[8] Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology 38 (1996) 275–280.

[9] Brinkkemper S., M. Saeki, F. Harmsen, Assembly Techniques for Method Engineering.Proc. of the 10th Conf. on Advanced Information Systems Engineering, Pisa Italy, 1998.

[10] M. Rossi and B. Ramesh and K. Lyytinen and J. Tolvanen {Managing evolutionary method engineering by method rationale } Journal of the association for information systems, 5(9), 2004, pp. 356-391.

[11] P.J. A ° gerfalk, K. Wistrand, Systems development method rationale: a conceptual framework for analysis, in: O. Camp, et al. (Eds.), Proceedings of Fifth International Conference on Enterprise Information Systems (ICEIS 2003), Angers, France, 23–26 April 2003,2003, pp. 185–190.

[12] J. Ralyte´, R. Denecke`re, C. Rolland, in: J. Eder, et al. (Eds.), Towards a generic model for situational method engineering, Proceedings of 15th International Conference on advanced Information Systems Engineering (CAiSE 2003), Klagenfurt, Austria, June 16–18, 2003, Springer, Heidelberg, 2003, pp. 95–110.

