Chapter IV

Exploring the Concept of Method Rationale: A Conceptual Tool to Understand Method Tailoring

Pär J. Ågerfalk, University of Limerick, Ireland
Brian Fitzgerald, University of Limerick, Ireland

ABSTRACT

Systems development methods are used to express and communicate knowledge about systems and software development processes, that is, methods encapsulate knowledge. Since methods encapsulate knowledge, they also encapsulate rationale. Rationale can, in this context, be understood as the reasons and arguments for particular method prescriptions. In this chapter, we show how the combination of two different aspects of method rationale can be used to shed some light on the communication and apprehension of methods in systems development, particularly in the context of tailoring of methods to suit particular development situations. This is done by clarifying how method rationale is present at three different levels of method existence. By mapping existing research on methods onto this model, we conclude the chapter by pointing at some research areas that deserve attention and where method rationale could be used as an important analytic tool.
INTRODUCTION

Systems development methods are used as a means to express and communicate knowledge about the systems/software development process. The idea is that methods encapsulate knowledge of good design practice so that developers can be more effective, efficient, and confident in their work. Despite this, it is a well known fact that many software organizations do not use methods at all (Iivari & Maansaari, 1998; Nandhakumar & Avison, 1999), and when methods are used they are not used literally “out of the box,” but are tailored to suit the particular development situation (Fitzgerald, Russo, & O’Kane, 2003). This tension between the method “as documented” (or as inter-subjectively agreed upon) and the method “in use” has been described as a “method usage tension” between “method-in-concept” and “method-in-action” (Lings & Lundell, 2004). This tension has given rise to an array of different approaches, ranging from contingency factor-driven method engineering (van Slooten & Hodes, 1996) through method tailoring and configuration (Cameron, 2002; Fitzgerald et al., 2003; Karlsson & Ågerfalk, 2004) to the various agile methods, such as XP (Beck, 2000) and SCRUM (Schwaber & Beedle, 2002).

A basic condition for a method to be accepted and used is that method users perceive it to be useful in their development practice (Riemenschneider, Hardgrave, & Davis, 2002). For someone to regard a piece of knowledge as valid and useful, the knowledge must be possible to rationalize, that is, the person needs to be able to make sense of it and incorporate it into his or her view of the world. Ethno-methodologists refer to this property of human behaviour as “accountability” (Dourish, 2001; Eriksén, 2002; Garfinkel, 1967); people require an account of the truth or usefulness of something in order to accept it as valid. This is particularly true in the case of method prescriptions since method users are supposed to use these as a basis for future actions, and thus use the method description as a partial account of their own actions. Hence, we follow Goldkuhl’s (1999) lead and use the term “action knowledge” to refer to the type of knowledge that is codified as method descriptions.

In order to better understand the rationalization of system development methods, the concept of method rationale has been suggested (Ågerfalk & Åhlgren, 1999; Ågerfalk & Wistrand, 2003; Oinas-Kukkonen, 1996; Rossi, Ramesh, Lytyinen, & Tolvanen, 2004). Method rationale concerns the reasons and arguments behind method prescriptions and why method users (e.g., systems developers) choose to follow or adapt a method in a particular way. This argumentative dimension is an important but often neglected aspect of systems development methods (Ågerfalk & Åhlgren, 1999; Ågerfalk & Wistrand, 2003; Rossi et al., 2004). One way of approaching method rationale is to think of it as an instance of “design rationale” (MacLean, Young, Bellotti, & Moran, 1991) that concerns the design of methods, rather than the design of computer systems (Rossi et al., 2004). This aspect of method rationale captures how a method may evolve and what options are considered during the design process, together with the argumentation leading to the final design (Rossi et al., 2004), and thus provides insights into the process dimension of method development. A complementary view on method rationale is based on the notion of purposeful-rational action. This aspect of method rationale focuses on the underlying goals and values that make people chose options rationally (Ågerfalk & Åhlgren, 1999; Ågerfalk & Wistrand, 2003) and provides an understanding of the overarching conceptual structure of a method’s underlying philosophy.
Maintaining Mappings between Conceptual Models and Relational Schemas
Yuan An, Xiaohua Hu and Il-Yeol Song (2010). *Journal of Database Management* (pp. 36-68).
www.igi-global.com/article/maintaining-mappings-between-conceptual-models/43729?camid=4v1a

On Non-Constrained, Constrained and Mandatory Many-to-Many Relationship Types
www.igi-global.com/article/non-constrained-constrained-mandatory-many/51113?camid=4v1a