ABSTRACT

Method configuration is a specific type of Method Engineering (ME) that takes an existing organization-wide Information Systems Development Method (ISDM) as its point of departure. Existing assembly-based ME approaches are not well suited to this task. As an alternative, this article suggests a metamethod approach to tailoring organization-wide ISDMs. We refer to this approach as the Method for Method Configuration (MMC). MMC takes into account the need to combine structure, which is one reason for choosing an organization-wide ISDM in the first place, with flexibility, which is essential for making the chosen ISDM fit actual projects. The metamethod is built using a three-layered reuse model comprising method components, configuration packages, and configuration templates. These concepts are combined efficiently to produce a situational method and thereby to facilitate the work of method engineers. [Article copies are available for purchase from InfoSci-on-Demand.com]

Keywords: IS Development Methodologies; IS Development Approaches; IS Project Management Methods and Tools; Software Development Methodologies; Systems Development Process

INTRODUCTION

Ineffective development practice is a perennial problem for many Information Systems (IS) development organizations (Brancheau, Janz, & Wetherbe, 1996; Riemenschneider, Hardgrave, & Davis, 2002; Roberts, Gibson, Fields, & Rainer, 1998). In response to this problem, many IS Development Methods (ISDMs) have been developed (Fitzgerald, Russo, & Stolterman, 2002; Roberts et al., 1998). There is clearly a need for structured and standardized ways of working in IS development. At the same time, there is a need for flexible work practices (Bajec, Vavpotič, & Krisper, 2006; Fitzgerald et al.,...
A standardized way of working makes projects effective and predictable and increases the likelihood of meeting deadlines, staying within budget constraints, and achieving a desired quality (Fitzgerald et al., 2002). Standardization of IS development practice occurs at three different levels. At the individual level, project members choose ISDMs and specific techniques for use in specific tasks to improve their personal and collaborative practice. At the project level, a wider agreement can be made concerning the use of a particular ISDM. Finally, at the organizational level, a specific ISDM can be chosen as the organization-wide “standard” method to be used in all projects. The expected benefits from implementing a standard ISDM across an organization include effective communication and reduced training costs due to a common modeling language, lowered maintenance costs, and the utilization of industry standards and existing tools. The search for organization-wide ISDMs has resulted in “off-the-shelf” ISDMs such as the Unified Process. Such ISDMs are often fully comprehensive, aimed at covering every aspect of IS development, and bundled with training, support, and various development tools from a specific vendor.

However, practical experience reveals that all projects are unique and require unique methodological support (van Slooten & Hodes, 1996). Some have also criticized organization-wide ISDMs, suggesting that they are too inflexible and hard to comprehend (Beck, 2000). The various parts of an ISDM are developed and included in the ISDM to minimize risks and to maximize the probability of success (Ropponen & Lyytinen, 2000). Because ISDMs aim to cover many different aspects of software practice, they will contain parts and address risks that are not of interest to every project. Hence, organization-wide ISDMs must be tailored to suit the particular development context (Fitzgerald, Russo, & O’Kane, 2003; Fitzgerald et al., 2002). To this end, different approaches have been suggested, ranging from formalized ones (Fitzgerald et al., 2003; Harmsen, 1997; Rolland, Prakash, & Benjamen, 1999; van Slooten & Hodes, 1996) to informal guidelines (Cameron, 2002; Cockburn, 2000). A rigorous approach to creating and tailoring ISDMs is that of Method(ology) Engineering, or ME (Brinkkemper, 1996; Kumar & Wellke, 1992). ME concerns methods for developing ISDMs, also known as “metamethods.” Traditionally, metamethods support the selection and integration of ISDM parts that together form consistent “situational methods.” This approach to situational ME is therefore assembly-based, whereby new methods are constructed for each new project. Such an approach does not take advantage of any existing organization-wide method.

In this article, we explore a different approach, called “method configuration” and defined as the planned and systematic adaptation of a specific method via the use of “reusable assets.” Here, a reusable asset is a pretailored part of an ISDM that fits a specific aspect of a development context. Method configuration is therefore a form of ME that explicitly utilizes an existing organization-wide ISDM. Furthermore, it promotes modularization to facilitate the exclusion, addition, and replacement of reusable ISDM parts. Specifically, we elaborate on the design and use of a Method for Method Configuration (MMC).

The article is organized as follows. The next section presents an overview of related research, followed by the research approach adopted. The following two sections are devoted to MMC. The first of these covers MMC’s core concepts, namely method components, configuration packages, and configuration templates. The second reports on empirical experiences from a number of action case studies of the use of MMC and provides an in-context perspective of the metamethod. A concluding discussion summarizes the article and points out implications for practice and research.
Collaboration in Virtual Worlds: Impact of Task Complexity on Team Trust and Satisfaction
www.igi-global.com/article/collaboration-in-virtual-worlds/195000?camid=4v1a

INDUSTRY AND PRACTICE: Object-Oriented Database: How Wide is the Application Range?
www.igi-global.com/article/industry-practice-object-oriented-database/51153?camid=4v1a

Real-Time Object-Oriented Database Support For Program Stock Trading
www.igi-global.com/article/real-time-object-oriented-database/51132?camid=4v1a