Representation of Situational Methods: Incorporating ISO/IEC 24744 into a Domain-Based Framework

Iris Reinhartz-Berger, Department of Information Systems, University of Haifa, Haifa, Israel

ABSTRACT

Method Engineering (ME) and Situational Method Engineering (SME) aim at providing effective solutions for building and supporting evolution of software and information systems development methods. For this purpose, method components are specified and composed into general-purpose development methods or situational methods, i.e., development methods that best fit the characteristics of a given project and its environment. Recently ISO/IEC 24744 has emerged for defining a metamodel and a notation for development methods. However, this standard lacks a systematic support for situational classification and maintenance. In this work, the authors suggest incorporating ISO/IEC 24744 into a domain-based framework, called Application-based Domain Modeling for Method Engineering (ADOM-ME), which supports specifying both general-purpose and situational methods in a single, simple, accessible, and scalable frame of reference. An exploratory study on the usability of ADOM-ME indicates that the approach can be utilized by information systems students that represent non-experienced method engineers and software developers.

Keywords: Development Methods, ISO/IEC 24744, Metamodeling, Method Engineering (ME), Methodology, Situational Method Engineering (SME)

INTRODUCTION

In the last decade, the need for effective, appropriate, and flexible development processes in information-based domains has increased. These domains are characterized by their intensive reliance on information management and processing, and include software, business, and systems engineering. The term ‘development method’ in these areas refers to (the specification of) the process to follow, the artifacts (or work products) to be used and generated, and the consideration of the people and the tools involved, during a development effort (Arthur & Verhoef, 1997; ISO, 2007).

Although development methods are one of the most significant key factors to the reduction of risks and the success of projects (Boehm, 1991), development methods usually cannot be used as they are and require different adaptations for supporting the specific needs of a given project properly (Domínguez & Zapata, 2007). For this purpose, method parts are analyzed and specified, rather than complete...
methods. These parts are differently termed by
the various approaches, e.g., road maps (van
Slooten & Brinkkemper, 1993), fragments
(Harmsen et al., 1994), chunks (Rolland et
al., 1998), patterns (Rolland et al., 2000),
method components (Mirbel & Ralyté, 2006),
and method services (Deneckere et al., 2008).
Although having slightly different meanings,
we refer in this work to all method parts as
method components, following Mirbel’s and
Ralyte’s observation that this is the most general
term (Mirbel & Ralyté, 2006). Having method
components analyzed and specified, they can
be composed to general-purpose development
methods (Brinkkemper, 1996) or to situational
methods, i.e., development methods that best
fit the characteristics of a given project and its
environment, e.g., the organization in which
the project is developed and the project’s client
(Harmsen et al., 1994; Mirbel & Ralyté, 2006).

Representation of method components and
development methods is important for docu-
mentation, usage, and maintenance purposes
in regular and situational method engineering.
Recently, ISO/IEC 24744 has emerged for
proposing both a metamodel and a notation for
representing different kinds of method compo-
nents (Henderson-Sellers & Gonzalez-Perez,
2008; ISO, 2007; ISO, 2010). This standard
provides concepts, symbols, and rules that are
relevant for development in information-based
domains. In particular, it refers to three model-
ing layers and its graphical notation includes
four diagram types.

Although ISO/IEC 24744 is an extensive
and recent standard, it lacks a systematic support
for situational classification and maintenance.
Such support is required for understanding the
situations to which each method component
is suitable and for creating “good” situational
methods. Furthermore, the richness of the nota-
tion (more than 30 symbols and four types of
diagrams) may cause problems in its utilization.
To overcome these limitations, we suggest
incorporating ISO/IEC 24744 metamodel to
a comprehensive domain-based framework,
called Application-based Domain Modeling
for Method Engineering (ADOM-ME). This
framework enables importing ISO/IEC 24744
as a domain (reference) model and guiding the
creation of valid method components, their
potential classification according to different
situational characteristics, and their composi-
tion to both general-purpose and situational
development methods. All these activities are
done within the same frame of reference, sup-
porting the creation of manageably scalable
and consistent models.

ADOM-ME is partially introduced and ex-
emplified in the work of Aharoni and Reinhartz-
Berger (2008), whereas their work from 2011
(Aharoni & Reinhartz-Berger, 2011) focuses
on the retrieval and composition capabilities
of ADOM-ME. Here we concentrate on the
representation capabilities of ADOM-ME and
analyze how they can improve ISO/IEC 24744
expressiveness and supply support for creating
situational methods. We further report on our
exploratory study on ADOM-ME usability.

The rest of the paper is organized as follows.
First, we provide the required background about
method engineering and situational method
engineering, including a brief review of the
terminology and the relevant approaches. We
further elaborate on ISO/IEC 24744 and its main
limitations in terms of supporting situational
methods representation. This is followed by a
section that presents and exemplifies ADOM-
ME, and a section that elaborates on the explor-
atory study on ADOM-ME usability. Finally, we
summarize and refer to future research plans.

METHOD ENGINEERING
AND SITUATIONAL
METHOD ENGINEERING

Terminology

Method components, which are the building
blocks of development methods, are in the basis
of both method engineering (ME) and situ-
ational method engineering (SME). They can
be adapted to particular development projects
utilizing different strategies, the most notable
of which are aggregation and specialization.
Related Content

The Impact of Regulatory Compliance on Agile Software Processes with a Focus on the FDA Guidelines for Medical Device Software
[www.irma-international.org/article/impact-regulatory-compliance-agile-software/53206/](www.irma-international.org/article/impact-regulatory-compliance-agile-software/53206/)

Designing Mobile Aspect-Oriented Software Architectures with Ambients
[www.irma-international.org/chapter/designing-mobile-aspect-oriented-software/66485/](www.irma-international.org/chapter/designing-mobile-aspect-oriented-software/66485/)

A Three Layered Approach for General Image Retrieval
[www.irma-international.org/chapter/three-layered-approach-general-image/27826/](www.irma-international.org/chapter/three-layered-approach-general-image/27826/)

Exploring the Perceived End-Product Quality in Software-Developing Organizations
[www.irma-international.org/article/exploring-perceived-end-product-quality/65560/](www.irma-international.org/article/exploring-perceived-end-product-quality/65560/)

Autonomic Business-Driven Dynamic Adaptation of Service-Oriented Systems and the WS-Policy4MASC Support for Such Adaptation
[www.irma-international.org/chapter/autonomic-business-driven-dynamic-adaptation/66797/](www.irma-international.org/chapter/autonomic-business-driven-dynamic-adaptation/66797/)