Developing Semantically-Enabled Families of Method-Oriented Architectures

Mohsen Asadi, School of Interactive Art & Technology (SIAT), Simon Fraser University, Surrey, BC, Canada

Bardia Mohabbati, School of Interactive Art & Technology (SIAT), Simon Fraser University, Surrey, BC, Canada

Dragan Gašević, School of Computing and Information Systems, Athabasca University, Athabasca, AB, Canada

Ebrahim Bagheri, School of Computing and Information Systems, Athabasca University, Athabasca, AB, Canada

Marek Hatala, School of Interactive Art & Technology (SIAT), Simon Fraser University, Surrey, BC, Canada

ABSTRACT

Method Engineering (ME) aims to improve software development methods by creating and proposing adaptation frameworks whereby methods are created to provide suitable matches with the requirements of the organization and address project concerns and fit specific situations. Therefore, methods are defined and modularized into components stored in method repositories. The assembly of appropriate methods depends on the particularities of each project, and rapid method construction is inevitable in the reuse and management of existing methods. The ME discipline aims at providing engineering capability for optimizing, reusing, and ensuring flexibility and adaptability of methods; there are three key research challenges which can be observed in the literature: 1) the lack of standards and tooling support for defining, publishing, discovering, and retrieving methods which are only locally used by their providers without been largely adapted by other organizations; 2) dynamic adaptation and assembly of methods with respect to imposed continuous changes or evolutions of the project lifecycle; and 3) variability management in software methods in order to enable rapid and effective construction, assembly and adaptation of existing methods with respect to particular situations. The authors propose semantically-enabled families of method-oriented architecture by applying service-oriented product line engineering principles and employing Semantic Web technologies.

Keywords: *Method Engineering, Method Oriented Architecture (MOA), Semantic Web, Software Development, Software Product Line*

DOI: 10.4018/jismd.2012100101

INTRODUCTION

The increase in the complexity of software-intensive systems has urged the integration of seminal approaches such as Object-Modeling Technique (OMT) and Objectory to form integrated (plan-driven) and unified frameworks such as the Rational Unified Process (RUP). Integrated approaches typically target the development of a vast variety of software applications, which increase the size of methods and make them become "cook-book" approaches. The recent critical literature reviews and comprehensive case studies have shown that such cook-book approaches do not work successfully in all circumstances (Rolland, 2009). Practitioners could potentially waste up to 35% of their efforts by following the steps of standard development methods (Harmsen, 1997). Moreover, the results of such studies reveal that the formal definition prescribed by a method in the forms of stages and steps widely differ from the method actually being used (Lings & Lundell, 2004). These issues have motivated the software engineering research community to establish the *Method Engineering (ME)* (Harmsen, 1997) discipline. The ME community concentrates on the idea of providing an "adaptation framework whereby methods are developed to match specific organization situation" (Rolland, 2009). The most prominent ME approach is *assembly-based method engineering* that creates a new method by assembling existing method components (Ralyte et al., 2003; Mirble & Ralyte, 2005). Despite the fact that ME has recently produced promising research results, there are still many open research challenges(Rolland, 2009). In this paper, we focus on the following two key challenges:

1. The lack of a standard model for describing method components limits the opportunities of method engineers, teams and organizations to share, discover, and retrieve distributed method components. When a method engineer wants to create a new method from scratch or by adapting (extending/constraining) an existing method, a common approach is to try reusing existing method components from method repositories. Therefore, method components need to be discovered and composed with other method components. Due to the lack of standards, method engineers are forced to reuse method components from the local proprietary repositories, without effective capabilities for retrieving method components in the repositories of their collaborators. In addition to this limitation of method component sharing, business opportunities of organizations are also limited. In fact, they cannot easily publicize and offer the methods that they are specialized in, as (for profit) services.
2. In essence, organizations initially adopt a method for the software development. Afterwards, components of the method may be subsequently added and gradually extended. Such extensions may be derived due to either the evolution of software development or various variations created for some specific method components. Some sources of these diversities may differ between domains of systems under development (i.e., desktop applications, web applications, and real-time systems) or newly emerging software development approaches such as Model Driven Development, Component based Software Development as well as method types (e.g., agile or plan-driven). Thus, there is the need for a systematic approach to manage variability of software methods and adapt software methods (families) that best suit the needs of a specific development context.

The first challenge has already been identified and discussed in the literature, and some researchers have proposed the use of SOA and Web service standards and principles in order to deal with this challenge (Rolland, 2009; Deneckere et al., 2008). To this end, the concept of *Method Services* was coined in analogy to the concept of services in SOA. Although the notion of method service provides a standard for