# A comparison of deontic matrices, maps and activity diagrams for the construction of situational methods

Valeria Seidita[1], Jolita Ralyté[2], Brian Henderson-Sellers[3], Massimo Cossentino[4] and Nicolas Arni-Bloch[2]

[1]DINFO, Università degli Studi di Palermo Viale delle Scienze, 90128 Palermo, Italy
[2]CUI, University of Geneva, Rue de Général Dufour, 24, CH-1211 Genève 4, Switzerland
[3]University of Technology, Sydney, PO Box 123, Broadway, NSW 2007, Australia
[4]SET - Université de Technologie Belfort-Montbéliard - 90010 Belfort cedex, France

**Abstract.** Several approaches have been proposed to support situational method engineering (SME), each of them providing different techniques and using different basic concepts. In this work, we propose a framework for comparing SME approaches based on a generic SME process model. Three approaches are presented and compared by using this framework.

## 1 Introduction

Situational method engineering (SME) involves, *inter alia,* a *method construction* element. Although there are many publications on the topic [5, 7, 8, 11], each offers its own individualistic approach. In this paper, we introduce an evaluative framework based on a generic situational method engineering process model. After a description of this process model in section 2, in the following section (3) we analyse, in turn, the use of three techniques (a) deontic matrices, (b) maps and (c) activity diagrams for their applicability to situational method engineering. In section 4, three approaches using these techniques are compared by applying our evaluation framework.

## 2 Constructing Situational Methods – A Generic Process Model

Starting from the assumption proposed by Gupta and Prakash [4] that a method engineering process is composed of three main phases, method requirements engineering, method design and method construction, we have described a high level (generic) process model for SME with the following phases: method requirements engineering, method fragments selection and method fragments assembly.

The first phase of SME aims to specify requirements for a project-specific method and can be decomposed into three main activities: project situation assessment, method/process goals identification and process model definition; together with affiliated tasks. The second phase encompasses the selection of method fragments/chunks from the repository corresponding to the requirements defined during the first phase. We identify three main activities in this phase: preliminary fragments selection, method fragments analysis and final selection. The third phase deals with selected method fragments/chunks assembly. It is refined into three activities: assembly technique assessment, final identification of process, and validation and evaluation.

We found it necessary to extend this framework by specifying 1) the kind of support provided by the application of each approach, 2) the presence of guidelines, 3) the possibility of using some kind of automation for each phase, 4) flexibility at three levels: *high* (each method fragment can be easily added to or removed from the process model), *medium* (every operation on fragments can be made under certain constraints) and *low* (no flexibility is supported).

## 3 Three Techniques for Method Construction

A *deontic matrix* is a two dimensional matrix the values in which serve to link the various process components. Deontic values for any specific pair of process components depend on the context of the specific project, the development team skills, etc. and are typical of the OPF approach (e.g. [3]). A process engineer has to configure OPEN by creating instances of its metamodel (i.e. method fragments) that are suitable for using on the specific project. In so doing, they have to use their own experience and knowledge on new method requirements.

A *Map* [9] is a navigational structure in the form of a graph where nodes are intentions and edges are strategies. It is possible to follow different strategies for each couple of target/source intentions, thus dynamically determining different solution paths between start and end. The Map formalism is used by Ralyté *et al.* [8] during the construction process of a new method by following three main steps: methods requirements specification, method chunks selection and method chunks assembly. A map is also used to represent the method chunk itself, thus enabling the designer to apply similarity measures between the requirement map and the method chunk representation to assess if a method chunk matches a specific requirement [7].

Three SME papers [1, 10, 11] use a version of UML *activity diagrams*, offering an approach for the development of a new method using typical steps of: (i) identifying the needs for the new method by analysing the application context; (ii) selecting, from existing methods, those meeting some required aspect; (iii) analysing selected methods and storing them in a method base; (iv) assembling method fragments into a new method to obtain situational methods.

Both [10, 11] claim to use a meta-modelling technique to model a complete process or part of it; however, the technique is in fact made up of two diagrams, a UML Activity diagram and a Class diagram, respectively used to model the process and its related concepts, resulting in a novel hybrid diagram named *process-data diagram*. The method engineer selects a set of existing methods that could fit the application context on the basis of personal knowledge and expertise, argued to be quite straightforward. The approach in [1] is more strictly based on SPEM's Activity Diagram [6]. While van de Weerd *et al.*'s approach [11] results in a joint diagram (activity plus class) to model process and data, SPEM presents these two views in a single diagram. Here, the process of creating a new method consists of analysing the new process and selecting and assembling the fragments. In this approach, SPEM activity diagrams are used only to model process and artefacts; the representation of data is not detailed and another diagram, where each artefact is related to the data it represents (according to a general meta-model of the system) [2], is used.

## 4 Comparison of the SME Approaches

Using the framework described in Section 2 we can evaluate the potentiality of each approach to support the construction of an SME process. For each approach we ask the following question: *Does it provide help for the activity carried out in each generic process phase?* In this sense we have to examine if, for each approach, the specific phase/activity/task is performed and how this is done. Table 1 presents the

**Table 1.** Properties in the comparison framework

| **Generic SME Process** | | | **SME Approaches** | | | |
|---|---|---|---|---|---|---|
| Pha-se | Activity | Task/Attribute | Map [8] | Deontic Matrices | Activity Diagram | |
| | | | | | [10, 11] | [1] |
| Method Requiremen | Project Situation Assessment | Characterisation of the project environment | S, SF, G, NT | S, I, NG, P | S, I, NG, NT | S, I, NG, NT |
| | | Identification of the project features | S, SF, G, NT | S, I, NG, P | S, I, NG, NT | S, I, NG, NT |
| | | Method situation evaluation | S, I, G, NT | S, I, NG, P | S, I, NG, NT | S, I, NG, NT |
| | Method/process goals identification | | S, SF, G, NT | S, I, G, P | S, I, NG, NT | S, I, NG, NT |
| | | Way of identification | Proc. | Proc. | Proc. | Proc, Prod |
| | Process model definition | | S, SF, G, NT | S, SF, G, NT | S, SF, NG, NT | S, SF, NG, NT |
| | | Source | {fs,br} | {fs} | {fs} | {fs,br} |
| | | Technique | All | constr. | constr. | constr. |
| Method Fragment | Preliminary fragments selection | | S, F, G, NT | S, F, G, T | S, I, NG, NT | S, F, G, T |
| | | Way of selection | Proc. | Proc. | Proc. | Proc, Prod |
| | Method fragments analysis | | S, F, G, NT | S, F, NG, T | NS | S, I, NG, NT |
| | Final selection | | S, F, G, NT | S, SF, G, T | S, SF, NG, NT | S, SF, NG, NT |
| Method Fragment | Assembly technique assessment | Positioning selected method fragments | S, SF, G, NT | NS | S, SF, G, NT | NS |
| | | Assembly technique | {Ass, Int} | NS | {Ass} | NS |
| | Final identification of process model | | S, SF G, NT | S, SF, G, T | S, SF, G, NT | S, SF, NG, NT |
| | Validation and evaluation | | S, SF, G, NT | NS | S, SF, G, NT | NS |
| | | Level of flexibility | High | High | High | High |
| Values: (1) S/NS supported/not supported; (2) I/SF/F informal/semi-formal/formal; (3) G/NG guidelines/no guidelines; (4) T/P/NT tool/prototype/no tool support; *fs* from scratch; *br* by reuse; *Proc* | | | | | | |

| Process-driven, *Prod* Product-driven; *Ass* Association; *Int* Integration |
|---|

comparison results using our evaluative framework. We can conclude that for the *Requirements Engineering phase* the three approaches do not present substantial differences; some have guidelines and/or specific tools and all of them result in a set of requirements that is the basis for the fragments selection phase. Both deontic matrices and maps provide more formal support in the *Fragment Selection phase*, reflected in the potential to use an automated tool for the selection. In contrast, the first approach based on activity diagrams is informal being based principally on the designer's knowledge of the repository or existing design processes. Only the last phase (final selection) prescribes the use of a semi formal diagram (activity diagrams) as a reference point for the final selection. In the *Fragment Assembly phase*, only the Map-based approach [8] formally supports the assessment of assembly techniques while all the others bring to a kind of assembly on the fly where fragments are selected principally based on designers' knowledge, data they deal with or on the results of applying deontic matrices, so it is almost obvious that they can be assembled by merely putting them together. The activity of validation and evaluation of the obtained method is supported in Ralyté's and van de Weerd *et al.* approaches, which provide specific quality validation rules.

In our future work we also aim to use this framework for evaluating other SME approaches and to investigate the possibility to combine different SME approaches.

## References

1. Cossentino M., Seidita V.: Composition of a New Process to Meet Agile Needs Using Method Engineering. Software Engineering for Large Multi-Agent Systems vol. III. LNCS 3390, Springer-Verlag, 2005.
2. Cossentino M., Sabatucci L., Seidita V.: SPEM Description of the PASSI Process. http://www.pa.icar.cnr.it/cossentino/FIPAmeth/metamodel.htm, 2003.
3. Firesmith D.G., Henderson-Sellers B.: The OPEN Process Framework. An Introduction, Addison-Wesley, pp.330, 2002.
4. Gupta D., Prakash N.: Engineering Methods from Method Requirements Specifications. Requirements Engineering Journal. 6(3), pp.135-160, 2001.
5. Henderson-Sellers B.: Process Metamodelling and Process Construction: Examples Using the OPEN Process Framework (OPF). Annals Software Engin. 14, pp.341–362, 2002.
6. OMG: Software Process Engineering Metamodel Specification, version 1.1. formal/05-01-06. Object Management Group, 2005.
7. Ralyté J., Rolland C.: An Assembly Process Model for Method Engineering. CAISE'01, Proceedings, LNCS 2068, Springer-Verlag, pp.267-283, 2001.
8. Ralyté J., Deneckère R., Rolland C.: Towards a Generic Method for Situational Method Engineering. CAiSE'03, Proc., Springer, LNCS 2681, pp. 95-110, 2003.
9. Rolland C., Prakash N., Benjamen A.: A Multi-model View of Process Modelling, Requirements Engineering J. 4(4), pp.169-187, 1999.
10. Saeki M.: Embedding Metrics into Information Systems Development Methods: An Application of Method Engineering Technique. CAiSE'03, Proceedings, LNCS 2681, Springer, pp.374-389, 2003.
11. van de Weerd I., Brinkkemper, S., Souer, J., Versendaal, J.: A Situational Implementation Method for Web-based Content Management System-applications: Method Engineering and Validation in Practice. Software Process: Improvement and Practice 11(5), 521-538, 2006.