Fundamentals and Challenges of Situational Method Engineering

Jolita Ralytė
jolita.ralyte@unige.ch
Method Engineering and its Evolution

Engineered Situated Method

Different Company / Project

Different engineering context & different method requirements

Universal Method

New application domains, new development principles

Require new methods

There is no one-size-fits-all methodology!

(Erik van Brinkkemper 1996)
Situational Method Engineering (SME)

Situational Method Engineering

is the discipline to build project-specific methods, called situational methods, from parts of the existing methods, called methods fragments.

(Brinkkemper 1996)

Motivations

Adaptability
to specific projects, companies, needs & new development settings

Reuse
of best practices, theories & tools

© J. Ralyté, CBI 2014
Meta-modelling - a technique to represent and formalise methods and method components

Reuse of method parts in construction of other methods

Modularity – a method is viewed as a collection of inter-related autonomous components

Flexibility in defining and applying methods
Overview of SME

Method Reengineering
For Reuse

Initial method description

Method Reengineering Guidelines and Metamodels

Modular Method Description

Method Base

Situational Method

Techniques for Situational Method Construction by reusing method parts

Method Engineering
By Reuse

Storage of method parts

Method Base

© J. Ralyté, CBI 2014
Key Method Engineering Artefacts

SME products
SME meta models & metamodeling languages to represent method parts


SME processes
SME strategies, approaches, workflows to combine, integrate, assemble method parts & to guide the SME process

A modularization issue based on two assumptions
1. A method includes two interrelated perspectives – process and product
2. Meta-modelling is an appropriate means to describe methods

• Describes how to construct a product (e.g. a schema) corresponding to the product model.
• The product is a result of the process model application.

Process Meta-modelling
Activity diagrams, BPMN, Map, ...

• A set of concepts and constraints to describe a system at different levels of abstraction.
• Semantics of concepts and rules to use them.

Product Meta-modelling
ER / UML class diagrams, OCL,...
SME Products

Different meta-modelling languages for method process and product knowledge
Types of Reusable Method Parts

(Henderson-Sellers et al. 2014)
Method Fragments

Separate modularization of process and product aspects
(Harmsen et al., 1994, Harmsen 1997, Brinkkemper et al. 1999)

- Product Fragments
- Process Fragments

Diagram:

- Method Fragment
- Process Fragment
- Product Fragment
- Conceptual Process Fragment
- Technical Process Fragment
- Conceptual Product Fragment
- Technical Product Fragment

Relationships:
- Requires: 0,m
- Produces: 0,m
- Consists of: 0,m
- Precedes: 0,m
- Is preceded by: 0,m
- Is part of: 0,1
- Supports: 1,m
- Is supported by: 0,m

© J. Ralyté, CBI 2014
Product and Process Fragments

Product fragment

Entity

Relationship

Attribute

is involved in involves

n

m

has belongs

Process fragment

Assess Documentation

data model available?

no

yes

Construct data model

no

yes

data model adequate?

no

Deliver data model

Modify data model

(Harmsen et al. 1994, Harmsen 1997, Brinkkemper et al. 1999)

© J. Ralyté, CBI 2014
OPF Method Fragments

Process, product and producer aspects defined separately
(Firesmith and Henderson-Sellers 2002)

- Work Product
- Work Unit
- Producer

→ ISO/IEC 24744
**Method Chunk**

**Chunk = Process + Product**

Tight coupling in a chunk of the process and related product parts

(Ralyte & Rolland 2001)
Example of Method Chunk

ChunkID: CH01, ChunkName: Use Case Model, Type: Aggregate; Origin: OOSE Method; … …

Reuse situation:
Application domain->Application type->All;
Application domain->Impact of legacy system->Functional domain reuse
Contingency factor->Innovation level->Business innovation
System engineering activity->Requirements elicitation; Requirements specification

Reuse intention: Specify functional system requirements following use case modelling strategy

Interface

<(Problem statement), Construct a use case model following OOSE method strategy>

Process part

Start

Elicit a use case
Actor-driven discovery

Scenario writing

Conceptualise a use case
Abstraction

Include

Stop
Completeness

Body

Product part

Use Case Model

1..*
initialises > 1..*
supports >
includes *
uses *

Actor
Name Definition

1..*

Use Case
Name Description

0..1

Basic Scenario

Exception Scenario

Scenario Description

Descriptor

© J. Ralytė, CBI 2014

(Ralyte & Rolland 2001)
Process-Deliverable Fragment

Explicit link between process and product parts

(van de Weerd et al., 2007)
Example of Process-Deliverable Fragment

(van de Weerd et al., 2007)
Including process and product perspectives in the same fragment  
(Cossentino et al. 2007)
Method Component

Including process and product perspectives in the same component

(Wistrand & Karlsson 2004; Karlsson & Wistrand, 2006)
Granularity of Method Parts

- **Granularity of fragments by Brinkkemper et al. (1998)**
  - **Method**: Information Engineering, UML
  - **Stage**: Requirements Analysis, Technical Design
  - **Model**: Data Model, UI model
  - **Diagram**: Object diagram, Activity diagram
  - **Concept**: State, Event, Identify States

- **Granularity in OPEN**:
  - **Work Unit**: Process -> Activity -> Task -> Technique

- **Granularity of method chunks**:
  - An aggregation mechanism supporting N levels of granularity
SME Process

Step 1: Situational method construction

- Construction Guidelines
- Situational Factors

Method Engineer
- uses
- selects
- constructs

Methodbase
- instances of classes in

Step 2: Method enactment

- Project Manager

Enacted Method

Metamodel
- conforms to

Tailored Methodology
- optional

M1

M2

© J. Ralyté, CBI 2014
Assembly-Based SME

Method Requirements Specification

- Situation Characterisation
- Method Goal
- Functional Requirements

Method Parts Selection

- Selection rules
- Similarity Measures
- Fitness Relationships

Method Parts Assembly

- Assembly techniques
- Assembly Operators
- Quality Rules

© J. Ralytė, CBI 2014
Reuse Frame - an hierarchy of criteria

✧ To specify project situation
✧ To specify method chunk reuse context – the descriptor

Characterising Project Situation

© J. Ralytė, CBI 2014
### Characterising Project Situation

**Contingency Factors**

*by (van Slooten & Hodes 1996)*

- Management commitment (for the IS project)
- Importance (of the project)
- Impact (of the project)
- Resistance and conflict
- Time pressure
- Shortage of human resources
- Shortage of means
- Formality (of the project procedures)
- Knowledge and experience
- Skills
- Size
- Relationships (between the IS under development and existing IS)
- Dependency (of the project to external factors)
- Clarity (of the project goals, etc)
- Stability (of the project goals)
- Level of innovation

**Contingency Factors**

*by Euromethod*

- Complexity: S (Simple), M (Moderate), C (Complex)
- Risk: L (Low), M (Moderate), H (High)
  - Applicable to target Domain and project Domain
- Project Features: Task, structure, actors and technology
- Target features: size, heterogeneity of actors, complexity of data, technology ...
Method Assembly Techniques

Method assembly strategies
(Ralyté & Rolland 2001)

Association strategy

Integration strategy

New added associations

Unified and merged common/similar elements
Assembly by Association of Product Fragments

Statechart

- Event
- Firing Condition
- Transition

Object Model

- Event
- Firing Condition
- Transition

- Consists of
- Belongs to
- Participates in

- State
- Post condition

- Service
- Association

- Object
- Class
- Attribute

new association

new property

new concept

(Brinkkemper et al., 1999)
Assembly by Association of Process Fragments

O1: Identify Objects and Classes
- List of Objects and Classes
  - O2: Identify Associations
    - Diagram with Classes and Associations
      - O3: Identify Attributes and Services
        - Object Model Diagram

S1: Identify States
- List of States
  - S2: Identify State changes and Triggers
    - State Transition Diagram
      - S3: Clustering States ...

OC1: Draw an Object Model (A)
OC2: Draw a Statechart (B)
OC3: Refine Statecharts
- Objectchart

Draw an Objectchart (C)

(Brinkkemper et al., 1999)
Uses Deontic Matrix to select method fragments:

- Identify process activities
- Link selected activities to the tasks (Activity-Task Deontic matrix)
- Link selected tasks to the techniques (Task-Technique Deontic matrix)

Example of Deontic Matrix linking Tasks and Techniques

(Nguyen and Henderson-Sellers, 2003)
## OPF Method Construction Process

Six steps of a possible top down method construction process
*(Nguyen and Henderson-Sellers, 2003)*

<table>
<thead>
<tr>
<th>Step</th>
<th>Deontic matrix to be used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select Activities</td>
</tr>
<tr>
<td>2</td>
<td>Select Tasks and allocate their usage to each selected Activity.</td>
</tr>
<tr>
<td>3</td>
<td>Select Techniques and allocate their usage to each selected Task.</td>
</tr>
<tr>
<td>4</td>
<td>Select Work Products and allocate them to the appropriate Task</td>
</tr>
<tr>
<td>5</td>
<td>Select Languages and allocate them for documenting each selected Work Product</td>
</tr>
<tr>
<td>6</td>
<td>Select Producers and allocate them for the performance of each selected Task.</td>
</tr>
</tbody>
</table>
Assembly by Integration of Method Chunks

Example
SME Goal: Use Case Model Extension with scenario writing guidelines
Process part integration
Assembly by Integration of Method Chunks

Example

SME Goal: Use Case Model Extension with scenario writing guidelines

Product part integration

C1: Use Case model

C2: L’Ecritoire model

Rename_Class (C1.Scenario, C1.Simple Scenario)
Rename_Class (C2.Scenario, C2.Structured Scenario)
ConnectVia_abstraction (C1.Simple Scenario, C2.Structured Scenario)
A generic taxonomy of assembly operators
(Ralyté et al. 2004)

Integration of various composition strategies – a generic SME process
(Ralyté et al. 2003)

Towards Common Ground in SME
- Modularization Constructs in Method Engineering
  (Agerfalk et al. 2007)
- Comparison of method fragments and method chunks
  (Henderson-Sellers et al. 2008)
- Ontology of Method Descriptors
  (Iacovelli & Souveyet 2011)

© J. Ralyté, CBI 2014
Remaining Issues in SME

- Modularization of method knowledge
  - Different meta-models
  - Different semantics
  - Different definition of granularity
- A poor understanding of the notion of situation
- A need for more syntactic and semantic validation techniques
- Too conventional workflow type of SME process modeling
- Lack of tools supporting SME
- Adoption of SME in practice
SME Trends and Challenges

- SME Usage perspective
  - From Engineering Issues to Usage Concerns

- Formalisation and reuse of method knowledge
  - From method components to method services and MOA

- Consideration of method evolution perspective
  - From static ME to evolutionary SME

- New situations and new requirements for ISD methods
  - From intra-disciplinary through inter-disciplinary to trans-disciplinary context of SME
A shift in focus: from Engineering Issues to Usage Concerns

To focus on:
- understanding corporate needs in terms of method deployment
- providing solutions that facilitate the usage of methods that proved to add value to engineers’ work
- making services dedicated to method usage easily available

Towards a creation of Community of Practice (CoP) (Mirbel 2007)

From method components to method services

© J. Ralyté, CBI 2014 (Rolland 2007, IFIP W.G. 8.1 ME'07)
No centralized method base but distributed method services available through the Internet

(Rolland 2007, IFIP W.G. 8.1 ME'07)
MaaS – Method as a Service

- Interfaces of method parts are more important to work out than the structure of the component knowledge
- Move to services instead of working exclusively on the semantic of method fragments and chunks
- Publication and accessibility become central issues

Using Web service technology to provide self-describing, platform agnostic method elements (MaaS), accessible through standard interfaces and that can be assembled together

(lacovelli et al. 2008, Deneckère et al. 2008)
Evolutionary Method Engineering

**Method Rationale** *(Rossi et al, 2004)*:
Trace of method changes & associated use experiences

**Method Configuration** *(Karlsson & Agerfalk, 2007)*:
Three-layered configuration reuse model consisting of method components, configuration packages & templates

**Incremental Method Engineering** *(van de Weerd et al. 2006, Mirandolle et al. 2011, Vlaanderen et al., 2011)*
A method adaptation by adding or removing method increments, in order to improve its overall performance and to increase its maturity level

**Method Families** *(Deneckère et al. 2011, Asadi et al. 2011)*
Organization of a set of method variants & configuration of the situation-specific method

© J. Ralyté, CBI 2014
Conclusion

Fundamentals of SME
- Method components/fragments/chunks and their repositories
- Situation-driven method construction approaches

Open Issues in SME
- Method quality validation
- Characterisation of situations
- Easiness of SME process

Trends and Challenges
- Usability (MaaS)
- Composability (MOA)
- Variability (Method families)
- Evolution & Agility
Suggestion for Summer Reading 😊