Method Engineering: Towards Methods as Services

Outline
- Backdrop
- MaaS

Prof. Colette Rolland
CRI
Université Paris1 Panthéon Sorbonne
Evolution in Information System Development Methods (ISD)

1. Universal Method
2. Engineered Situated Method

Method Engineering
Method engineering is the discipline of developing, customizing, and/or configuring a situation-specific method from parts of existing methods.

[Brinkkemper 96, Leppanen 2006]
Method Engineering

Motivations
Adaptability
(to specific projects, companies, needs & new development settings)

Reuse
(of best practices, theories & tools)
Method Engineering

ME process & research questions

Initial Method Description

Reverse engineering step

Method reengineering guidelines

Situational Method

Construction of a new method reusing method parts

Several strategies for SME

Method Base

Storage of method parts in a method base

Modular Method Description: Modular Method model

ICSP 2008, Leipzig
# Method Engineering

*Software engineering vs Method Engineering*

<table>
<thead>
<tr>
<th>Software engineering</th>
<th>Method engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modelling (specification)</td>
<td>Meta- Modelling</td>
</tr>
<tr>
<td>CASE</td>
<td>CAME</td>
</tr>
<tr>
<td>(Computer Aided S/W Engineering)</td>
<td>(Computer Aided Method Engineering)</td>
</tr>
<tr>
<td>Software Base</td>
<td>Method Base</td>
</tr>
<tr>
<td>Software metrics</td>
<td>Method metrics</td>
</tr>
</tbody>
</table>

*Motoshi Saeki talk (EMISE)*
Method Engineering

Key Method Engineering Artefacts*

ME products

ME meta models & meta-modelling languages to represent method parts

ME processes

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

Harmsen 97, Heym & Osterle 92, Venable 93, Kelly et al 96, Graham et al 97, OMG 05, Prakash 99, Ralyte 01, * Leppanen 06

Kumar & Welke, 1992, Oei, 95, Phlihon et al 97, Heym & Osterle 97, Rolland & Prakash 96, Saeki, 98, Leppanen 00, Saeki 03, Karlsson & Agerfalk 04, * Leppanen 06

ICSP 2008, Leipzig
A modularization issue based on two assumptions

1. A method is composed of a product model and a process model
(2) Meta-Modelling is an appropriate means to describe methods (Yourdon & Coad)

Concept = {Class, Attribute, Service}
Relationship = {has(CS), has(CA)}

(a) Meta model of product part

procedure = {identify classes & objects, identify attributes}

(b) Meta model of process part

*Motoshi Saeki talk (EMISE)
The Method Fragment Perspective

[Harmsen 94, Harmsen 97, Brinkkemper 99]

**Product Fragment**

- Entity
- Relationship
- Attribute

**Process Fragment**

- Assess Documentation
- Construct data model
- Inspect data model
- Deliver data model
- Modify data model

Data model available? yes no
Data model adequate? yes no
The Method Chunk Perspective


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

<(Problem description), Discover goal / scenario couples with CREWS-L’Ecritoire strategy>
Method Engineering Products

Towards a consensual view (Cossentino et al., 2006)

Method fragments (Brinkkemper et al)
Method chunks (Ralyté et al)
OFP (Henderson Sellers et al)
FIPA fragments (Cossentino et al)
Method Engineering Products

♦ Using philosophical foundations

- Intentionality is at the core of cognition (all modern philosophers: Brentano, Twardowski, Husserl, Sartre, Merleau Ponty, Berner,…)

- Intention is a mental state that integrates desires and beliefs and determines actions

Intentionality Model to Support Process Guidance

Intentional action attributes

- Desire
- Belief
- Intention
- Skill
- Awareness

Intention baseline
Method Engineering Products

♩ Guiding the achievement of intentions

Intention refinement

Nature meta-model

Context

Intention

motivates

implies

raises

modifies

Situation

Action

Product

ICSP 2008, Leipzig
Method Engineering Products

Choice Guideline in NATURE

Possible choices

CC1: a2
CC1: a3
CC2: a4 or a9
or a10 or a11
CC3: a1 or a7 or a8
CC4: a6
CC5: a5
CC6: a1 or a7
CC7: a8

a1: cardinalities are: <t, s, p - p, s, v>
a2: cardinalities are: <p, ?, ? - ?, ?, ?>
a3: cardinalities are: <?, ?, ? - p, ?, ?>
a4: cardinalities are: <t, m, p - t, s, p>
a5: cardinalities are: <?, ?, v - ?, ?, ?>
a6: cardinalities are: <?, ?, p - ?, ?, v>
a7: cardinalities are: <t, s, p - p, s, p>
a8: cardinalities are: <p, m, v - p, m, v>
a9: cardinalities are: <t, m, v - t, s, p>
a10: cardinalities are: <t, s, t - t, s, p>
a11: cardinalities are: <t, s, v - t, s, p>
Method Engineering Products

Less consensual view on method part relationship types

- Aggregation mechanism (road maps, trees of contexts, etc.)

Variability (choice context)

- Variant
- Mandatory

Generic patterns (Plihon96)

- Abstracts
- Composed of

Method Part

- Primitive
- Aggregate
Method Engineering Processes

*Emphasis on composition strategies* (classification by Ralyté & Rolland)

- **Assembly based**

- **Extension based**

- **Paradigm based**

*Motoshi Saeki talk in EMISE*
Method Engineering Processes

♣ Method Assembly* : classification by J.Ralyté & C.Rolland

Integration strategy

Association strategy

New added association

Overlapped & unified
Method Engineering Processes

_method Assembly by association*: example of product models assembly

Object Model

Class

has

has

has

Service

Attribute

State Transition Model

ObjectChart, Coleman et al92
Method Engineering Processes

- **Method Assembly by integration: L’Ecritoire and OOSE**

  - **Product assembly operators**
    - RENAME (concept, link, property)
    - ADD (concept, link, property)
    - DELETE (concept, link, property)
    - GENERALISE (concept)
    - OBJECTIFY (link, property)
    - MERGE (concept, link, property)

  - **Process assembly operators**
    - RENAME (intention, section)
    - ADD (intention, section)
    - DELETE(intention, section)
    - MERGE(intention, section)
**Method Engineering Processes**

Less studies on ME processes
“Towards a life cycle for method engineering “(Gupta, Prakash 2007)

<table>
<thead>
<tr>
<th>Stage</th>
<th>Process</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements engineering</td>
<td>Intention matching</td>
<td>Goal of the method To-Be</td>
<td>Intentionally similar methods</td>
</tr>
<tr>
<td>Design engineering</td>
<td>Architecture matching</td>
<td>Architectures of intentionally similar methods</td>
<td>Architecturally similar method</td>
</tr>
<tr>
<td>Construction engineering</td>
<td>Organization matching</td>
<td>Workflows of architecturally similar methods</td>
<td>Method To-Be</td>
</tr>
</tbody>
</table>

Bajec et al 2007, Agerflak 2007, Mirbel & Ralyté 2006 etc.
Method Engineering Processes

A generic process model for method engineering (Ralyté, Rolland 2003)

Two key tasks to perform:
- Set method engineering goal
- Construct a method that matches this goal

A number of strategies

Method Engineering Processes

- Start
- Set Method Engineering Goal
  - Method adaptation-based strategy
  - From scratch strategy
  - Comparison strategy
- Construct a Method
  - Paradigm-based
  - Extension-based
  - Assembly-based
- Evaluation strategy
- Stop
- Completeness strategy

Two key tasks to perform:
- Set method engineering goal
- Construct a method that matches this goal

A number of strategies
Method Engineering Processes

- A still fragmented position on designing ME processes (Leppanen 2006)

Goodies …….

- a rich set of composition approaches and procedures
- attempts to integrate various composition strategies
- set of generic taxonomy of assembly operators
- proposals for decomposing ME into ME workflows
Method Engineering Processes

and remaining issues …….

- need for a better understanding of the dimensions of situational method development (Aydin et al 2007)
- poor understanding of the notion of situation (Bucher et al, 2007)
- need for more syntactic and semantic checking techniques
- more advanced retrieval techniques
- too conventional workflow type of ME process modeling
Method Engineering

 pobl  Where to go next?
Method Engineering : MaaS

♣♣ A shift of focus : from engineering issues to usage concerns

From components to services
Towards CoP (Mirbel 07)
Standards (ISO/IEC24744)

- Emphasize and standardize MP Interface descriptions
- Publish and make them publicly available
- Provide contextual information to ease MP finding
- Facilitate MP composition
Method Engineering : MaaS

Towards MOA : Method Oriented Architecture

- Use method services as fundamental elements
- Reorganize a portfolio of existing methods into self-describing, elements (services), accessible through standard interfaces and that can be assembled together
- Based on an interaction between three kinds of method agents

Diagram:
- Publish
- Method Provider
- Find
- Method Registry
- Bind
- Method Client
Method Engineering : MaaS

MaaS : Methods as Services

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

Types:
- LastTradePriceRequest (ticketSymbol: String)
- LastTradePrice (price: float)

Messages:
- GetLastTradePriceInput (body: LastTradePriceRequest)
- GetLastTradePriceOutput (body: LastTradePrice)

Port-type:
- StockQuotePortType GetLastTradePrice (in: GetLastTradePriceInput, out: GetLastTradePriceOutput)

Binding:
- StockQuoteSoapBinding (StockQuotePortType, soap, document, http) GetLastTradePrice : http://example.com/GetLastTradePrice

Web Service

StockQuoteService

GetLastTradePrice (LastTradePriceRequest) -> (LastTradePrice)

http://example.com/stockquote
Method Engineering : MaaS

MaaS : Dealing with complex types (schemas) & their associated meta-models

Need to understand the meta-model

Method Engineering : MaaS

Method Service

ImproveRoleService

ImproveRoleAction (inputSchema, roleA, C, D) -> (resultSchema)
Method Engineering : MaaS

♣♣ MaaS : Schemas as XMI documents & MetaModels as XMI DTD

- MOF
- MetaModel (ex UML)
- Schema (ex UML schema)
- Instances (ex Object schema)

- represented by
- compliantTo
- instanceOf
MaaS : An XMI based solution

Types:
- ImproveRoleRequest(inputSchema:XMIDocument, RoleName: String, ClassWithRole: String, ClassWithoutRole: String)
- ImproveRoleResult(ResultSchema:XMIDocument)

Messages:
- ImproveRoleInput(body:ImproveRoleRequest)
- ImproveRoleOutput(body:ImproveRoleResult)

Port-type:
- ImproveRolePortType ImproveRoleAction
  (in: ImproveRoleInput out: ImproveRoleOutput)

Binding:
- ImproveSoapBinding(ImproveRolePortType, soap, document, http)
  ImproveRoleAction:
  http://maasexample.crinfo.univ-paris1.fr/ImproveRoleAction

http://maasexample.crinfo.univ-paris1.fr/improverole
Method Engineering: MaaS

Platform independent & accessible through the Internet

Method Engineering: MaaS

Method
Provider

Method
Registry

Method
Client

Publish

Invoke

Retrieve

WS Java

WS C#

WS Python

WSDL

WSDL

UDDI

UDDI

Win Vista

Linux

Os X

XMI
From ME to MaaS Management

- **Actions**
  - performs
  - publishes
  - uses
  - becomes

- **Service provider**
- **Market-maker**
- **Service operator**
- **Service client**
- **Service aggregator**

- **Managed services**
- **Composite services**
- **Basic services**

- **Description & Basic Operations**
  - Capability
    - Interface
    - Behavior
    - QoS
  - Publication
    - Discovery
    - Selection
    - Binding

- **Operations**
  - Assurance
  - Support

- **Composition**
  - Coordination
  - Conformance
  - Monitoring
  - QoS

- **Market**
  - Certification
  - Rating
  - SLAs
From ME to MaaS Management

Some other issues

- Enriching MaaS semantics
  - Semantic annotation of method services
  - Method Ontology

- Providing meta-support as services

- Increasing interaction capabilities in services & meta-services
Method Engineering : MaaS

Dealing with interactive advices

Using the WSRP (Web Service Remote Portlet) technology to combine interface & functional services in a single method service.
Method Engineering: Towards Methods as Services

Conclusion

Motivations

Composability

Dynamicity

QoS

Variability

Usability

(easy use of method services)

Needed assembly of elements (method services)
Thank you for your attention