Introduction to Method Engineering

Session 1
February 8, 2010
Suppose...

• You just graduated from MBI and got a nice job at Cap Gemini

• Your first project: Implement an Enterprise Content Management system in a large organization

• Where do you start?

• What activities are you planning to perform?
Which one to choose?
And most methods, techniques and models are related...

But what are the differences?
This course will help you in...

• Providing of insight and skills into the systematic description, explanation and evaluation of all aspects of the methodology of ICT systems development

• Being able to use and adapt methods, techniques and tools for the situation of the project or system

• Contributing to the body of knowledge on method engineering

• Being able to review scientific literature and analyzing a paper in depth
Lecturers

- Inge van de Weerd
- Sjaak Brinkkemper

Student assistants

- Peter van Stijn
- Themistoklis Karavellas
Agenda

• Introduction to method engineering
• Practical issues
• Assignment week 6 & 7
Typical issues in the Information Systems domain

- How should this project be executed?
- Which method is the best?
- What model should be created now?
- What is a good tool for that method?
- How can we get to the next level of process maturity?
Response to these issues

• No generic answer can be given.
• It all depends on the situation:
  – project, company, people, budget, timing, technology, platforms

What is needed:
• Formal study of methods and tools
• Adaptation of methods

→ Method Engineering
Method engineering

Studies in Method Engineering:
• Formal description of methods
• Identification of method fragments
• Method comparison
• Incremental method evolution
• Formalization of method properties
• Derivation of tool support
• Situational configuration of project specific methods
Comparison Method Engineering and Software Engineering

Method Engineering

- Designer of Java (J. Gosling)
- Designer of ERD (P. Chen)
- Programming language (e.g. Java)
- Modeling language (e.g. ERD)

Software Engineering

- A Java programmer
- A data analyst
- a Java program myHelloWorld.java
- An ERD diagram, e.g. on students/courses

Based on slides Metamodeling for Method Engineering, (c) 2009 M. Jeusfeld
Main concepts in Method Engineering

• Method
• Methodology

• Technique
• Notation
• Procedure

• Model
• Meta-model
• Diagram

• Tool
• MetaCASE

Read:
Method

Definition
A method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products.

Origins from Greek “methodos”: way of investigation

Examples
RUP, UM, SSADM, SDM, ITIL, T-map, DSDM, DYA, Scrum, and many, many others.
Examples of online methods

• DSDM

• Open Process Framework
  http://www.opvro.org/index.html

• Software Product Management
  http://www.softwareproductmanagement.org/

• SCRUM
Methodology

Definition
The methodology of systems development is the systematic description, explanation and evaluation of all aspects of methodical systems development.

or: The Science of Methods

Often methodology is used where method is meant.

Methodological schools
- Software engineering
- Information systems (IS)
- Socio-technical approaches
- Agile movement

So: Scrum and XP are methods in the methodological school of the Agile movement
Main concepts in Method Engineering

- Method
- Methodology
- Technique
- Notation
- Procedure
- Model
- Meta-model
- Diagram
- Tool
- MetaCASE
Technique

Definition
A technique is a procedure, possibly with a prescribed notation, to perform a development activity

Examples
– data modelling with entity-relationship diagrams
– interviewing with plain natural language
– requirements modelling with use-case diagrams
– pseudo-coding with Nassi-Schneiderman diagrams
Notation and procedure

Definitions:

A **notation** is a system of symbols with a corresponding set of rules to construct artefacts used in communication.

A **procedure** is a stepwise description of a process.

Examples of notation:

- UML
- English language
- Music score
- Predicate logic
Interviewing

Interview process
1. Establish information needs.
2. Prepare for the interview.
3. Plan your questions.
4. Be careful about taking notes.
5. Know your subject.
6. Be specific.
7. Be accurate.
8. Look for color.
9. Do not talk too much.
10. Remember your sense of humor.

Integrated Publishing (www.tpub.com)
Pseudocoding

Technique

Notation

(Used for designing complex software programs)
Classification of techniques

• **Formality of the notation**
  – natural language
  – structured graphics
  – mathematical

• **Type of development activity, e.g.:**
  – data modelling
  – process modelling
  – interaction design
Main concepts in Method Engineering

- Method
- Methodology
- Technique
- Notation
- Procedure
- Model
- Meta-model
- Diagram
- Tool
- MetaCASE
Model

Definition:
A model is a representation that contains statements about the properties of an artifact (object) of a real or imagined world (universe of discourse).

(Jeusfeld, 2009)

Models can be expressed in diagrams.

Examples
• Maquette (scale model)
• Business model (a framework of the business logic of a firm)
• Data model (a description of database structure)
Meta-model

Definition
A meta-model is a model that consists of formal statements about models.

Strictly speaking, a meta-model is also a model but its universe of discourse is the set of models that are described by the meta-model.

(Jeusfeld, 2009)

Meta-models can be expressed in diagrams.

Examples
• Meta-model of an entity-relationship diagram
• Meta-model of a class diagram
Entity-relationship diagram

Model of the structure of a database, expressed in an ERD

- Used for modeling data for information systems
- Symbols:
  - entity
  - relationship
  - attribute
Class diagram

- Used for data modeling and for functions of classes

Model of the structure and interaction of an application, expressed in a class diagram
Use case diagram

- Used to represent functionality provided by a system in terms of actors and dependencies.
Four levels of reasoning

M0: Data level

M1: Model level

M2: Meta-model level

M3: Meta-meta-model level

Or: language definition level (c.f. Jeusfeld 2009)

Or: language level (c.f. Jeusfeld 2009)

See also: Object Management Group OMG/MOF Meta Object Facility (MOF) Specification
Abstractions in statements about artifacts

**Meta-model level**
“subject – predicate – object”

**Model level**
“A lecturer teaches a course.”
“A student follows a course.”

**Data level**
“Inge teaches Method Engineering.”
“Peter follows e-Business.”

Based on slides Metamodelling for Method Engineering, (c) 2009 M. Jeusfeld
Main concepts in Method Engineering

- Method
- Methodology
- Technique
- Notation
- Procedure
- Model
- Meta-model
- Diagram
- Tool
- MetaCASE
**Tool**

**Definition**
A tool is a possibly *automated means* to support (a part of) the development process

- Support a collection of different notations,
- Assistance to whole development life-cycle
  - CASE tool
  - Integrated Project Support Environments (IPSE)
  - Analysts Workbenches, Project Management Workbench

**Examples**
COOL:Gen, Rational Rose, TogetherSoft, Visio, BeWise, Flower, and many more
SmartUML
MetaCASE

Design your modeling language

Use your modeling language

State name

DisplayFn

StopTime

startTime

sysTime

running

Stopped

Up

A

Down

A

startTime

stopTime

sysTime

Blinking:

In this state the Stopwatch is stopped and current stop time is shown on the display.
BREAK
Discussion

• What are your experiences with applying
  – methods,
  – techniques, or
  – tools?

• Do they work in all situations?
  – When do they?
  – When don’t they?
Observations

- **Different IT systems**
  - Information systems
  - Web-applications, portals
  - Workflow, business intelligence
  - Product software, commercial software
  - Real-time systems, embedded software

- **Different domains**
  - Financial, insurance, banking
  - Educational
  - Government, public sector
  - Transport, logistics
  - IT services
  - Manufacturing, production
  - Service industry
  - Energy, oil, utilities

- **Different platforms**
  - Relational DBMS
  - Java, J2EE
  - MS Access, .NET,
  - LAMP (Linux, Apache, MySQL, PHP)
Motivation

- There is not one method that fits all
  - Tolvanen (2000): 65% of organizations which use methods develop them in-house and 85% of organizations adapt existing methods in-house

- Understand how to extend or adapt methods and techniques to
  - the current project
  - future project
  - the organisation
  - the platform or application
Method Engineering

Definition:
Method engineering is the engineering discipline to design, construct, and adapt methods, techniques and tools for the systems development.

c.f. Software Engineering
Situational method

Definition
A situational method is a method tuned to a specific situation.

Situational Method Engineering is the area of method engineering focusing on situational methods.
Assembly-based

- Strategy based on the reuse of method components extracted from existing method and stored in some method base.

This technique helps to select and assemble different method components in order to construct a new method.

(Ralyté, Deneckère & Rolland, 2003)
Assembly-based situational method engineering

1. Project Characterisation
2. Selection and Assembly of Method fragments
3. Method Description
4. Method base
5. Situational Method
Meta-modeling

Problem
How to describe and store methods in the method base, in order to easily reuse the method fragments in the method engineering process?

Solution
Meta-modeling technique to describe both process and deliverable part of the method
→ Process-Deliverable Diagram (PDD)

Advantages
Clear, compact, consistent with UML standards.
Process-deliverable diagram example:
Risk workflow in UML-Based Web Engineering (Koch, 2000)
Example (2)

Combination of methods

GX method
Unified Process
UML-based Web Engineering

EXTENSIVE REQUIREMENTS ELICITATION

GOALSETTING

BACKGROUND
FEATURE LIST
ASSUMPTION
ASSUMPTION
SCOPE

DOMAIN MODEL

APPLICATION MODEL

DESCRIPTION

SOFTWARE REQUIREMENTS

USE CASE MODEL

ACTOR
USE CASE

REQUIREMENTS DOCUMENT

DEFINITION OF RELATIONS

TERM
RELATION

CLASS DIAGRAM

APPLICATION IMPLICATIONS

USER INTERFACE
NAVIGATION
MIGRATION ISSUE

ADDITIONAL REQUIREMENT

EXTENSIVE REQUIREMENTS ELICITATION

GOALSETTING

BACKGROUND
FEATURE LIST
ASSUMPTION
ASSUMPTION
SCOPE

DOMAIN MODEL

APPLICATION MODEL

DESCRIPTION

SOFTWARE REQUIREMENTS

USE CASE MODEL

ACTOR
USE CASE

REQUIREMENTS DOCUMENT

DEFINITION OF RELATIONS

TERM
RELATION

CLASS DIAGRAM

APPLICATION IMPLICATIONS

USER INTERFACE
NAVIGATION
MIGRATION ISSUE

ADDITIONAL REQUIREMENT

Requirements validation
Example (3)

Combination of methods
Agenda

• Introduction to method engineering
• Practical issues
• Assignment week 6 & 7
Course outline: schedule

• Lectures
  – Tuesday 11.00 - 12.45
  – Friday 15.15 - 17.00 (except last three weeks)

• Workshops
  – on Friday 13.15 – 15.00

• Midterm exam: Tuesday, March 1
• Final exam: Tuesday, April 20

• Check http://www.cs.uu.nl/docs/vakken/me/
## Planning

<table>
<thead>
<tr>
<th>Wk</th>
<th>L/W</th>
<th>Date</th>
<th>Subject</th>
<th>Speaker(s)</th>
<th>Deadlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>L</td>
<td>Feb 8</td>
<td>Introduction to Method Engineering</td>
<td>Sjaak Brinkkemper &amp; Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>Feb 11</td>
<td>Method description</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>Feb 11</td>
<td>Assignment explanation</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>L</td>
<td>Feb 15</td>
<td>Meta-data modeling</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>Feb 18</td>
<td>Method description</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>Feb 18</td>
<td>Meta-process modeling</td>
<td>Inge van de Weerd</td>
<td>February 18 (15.00h): Method description</td>
</tr>
<tr>
<td>8</td>
<td>L</td>
<td>Feb 22</td>
<td>Process-deliverable diagrams; ME patterns</td>
<td>Sjaak Brinkkemper</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>Feb 25</td>
<td>Meta-modeling</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td></td>
<td>Overview of method engineering research &amp; Method comparison</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>L</td>
<td>March 1</td>
<td>Mid-term exam</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>March 4</td>
<td>Meta-modeling</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>March 4</td>
<td>Incremental method evolution</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>L</td>
<td>March 8</td>
<td>Method assembly</td>
<td>Sjaak Brinkkemper</td>
<td>March 11 (15.00h): Process-deliverable diagram</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>March 11</td>
<td>Meta-modeling</td>
<td>Henk Verschuur</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>March 11</td>
<td>TBD</td>
<td>Henk Verschuur</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>No lectures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td>March 22</td>
<td>Meta-modeling in the Online Method Engine</td>
<td>Inge van de Weerd &amp; Kevin Vlaanderen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>March 25</td>
<td>Draft paper</td>
<td>Inge van de Weerd &amp; Kevin Vlaanderen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>March 25</td>
<td>Applications of ME in master thesis research</td>
<td>Inge van de Weerd &amp; Kevin Vlaanderen</td>
<td>March 25 (15.00h): Draft paper</td>
</tr>
<tr>
<td>13</td>
<td>L</td>
<td>March 29</td>
<td>Situational method engineering &amp; the method association approach</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>April 1</td>
<td>Presentations</td>
<td>Inge van de Weerd</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>L</td>
<td>April 5</td>
<td>Meta-modeling with ConceptBase</td>
<td>Manfred Jeusfeld</td>
<td>April 8 (17.00h): Peer review</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>April 8</td>
<td>Presentations</td>
<td>Manfred Jeusfeld</td>
<td></td>
</tr>
</tbody>
</table>
Attendance

- Attendance to lectures and workshops is strongly advised, but not compulsory.
- Exceptions: the guest lectures and presentation workshops are compulsory.
Course outline: contents

• Reading of scientific literature
  – Can be downloaded from BSCW

• Writing a paper on a workshop topic
  – Methods, approaches and techniques in various information and computer science areas, mainly focused on software product management

• Presentations and peer reviews by students (week 13 & 14)
Examination

• Deliverable grade: 40%

• Midterm exam: 25%
  – On meta-modeling

• Final exam: 35%
  – Lecture and workshop materials including notes are subject of this exam.

• All grades should be 5.0 or higher
• In week 22 a re-exam is scheduled. You can choose which exam you want to retake, but you can only retake one.
Deliverable grade

- Method description (20%)
- Process-deliverable diagram (20%)
- Draft paper
- Presentation (10%)
- Peer review
- Final paper (40%)
- Wiki entry (10%)
How to choose a topic?

• Topics will be published on Wednesday, February 9, 17.00h, on the course website: http://www.cs.uu.nl/docs/vakken/me.

• Who comes first can choose first.
For more inspiration and to find related literature

- [http://www.cs.uu.nl/wiki/bin/view/MethodEngineering](http://www.cs.uu.nl/wiki/bin/view/MethodEngineering)
- [http://www.softwareproductmanagement.org/](http://www.softwareproductmanagement.org/)
  - Click ‘Competence model’
Agenda

- Introduction to method engineering
- Practical issues
- Assignment week 6 & 7
Assignment week 6 & 7: Method description

- Write the description of your chosen method
  - Introduction (global overview, creators method)
  - Example (preferably your own example)
  - Related literature (origins, positioning, application)
  - References (use APA style, see ‘Resources’ on website)

- Summarize, cite, recap, mention and review other work.

  (or something very terrible will happen to you)
Submitting your assignment

• Deadline: Friday, February 18, 15.00h

• Submit your .doc or .pdf to:
  – Ephorus
  – Twiki
  – A hardcopy in my mailbox
How to start?

- Read this week’s paper (Brinkkemper, 1996)
- Check assignments from earlier years

- On Wednesday 17.00, we publish the list of topics and corresponding papers in a Google spreadsheet. Check the topics, scan the papers, and choose the topic you prefer by writing your name next to the topic.

- Friday:
  - Workshop: working on method engineering assignment
    - Read the paper, find related literature, start writing the method description
  - Lecture: Further explanation on the assignment
    - How to find literature, how to make proper references, how to make an examples, etc.
QUESTIONS?
References

