Overview of ME research

Session 6a
25 February 2019

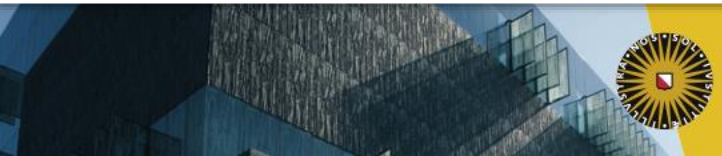
Prof.dr. Sjaak Brinkkemper
Dr. Sietse Overbeek

Universiteit Utrecht

Agenda

- We are standing on the shoulders of giants…
  - Method engineering
  - Creation of methods
  - Tool support
  - Method comparison

- …but do not always agree with each other



Google scholar beta

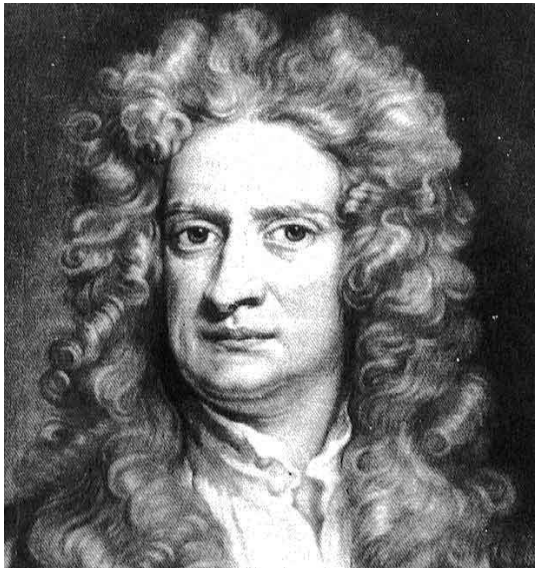
Search Advanced Scholar Search

Articles (include patents) Legal opinions and journals

Stand on the shoulders of giants

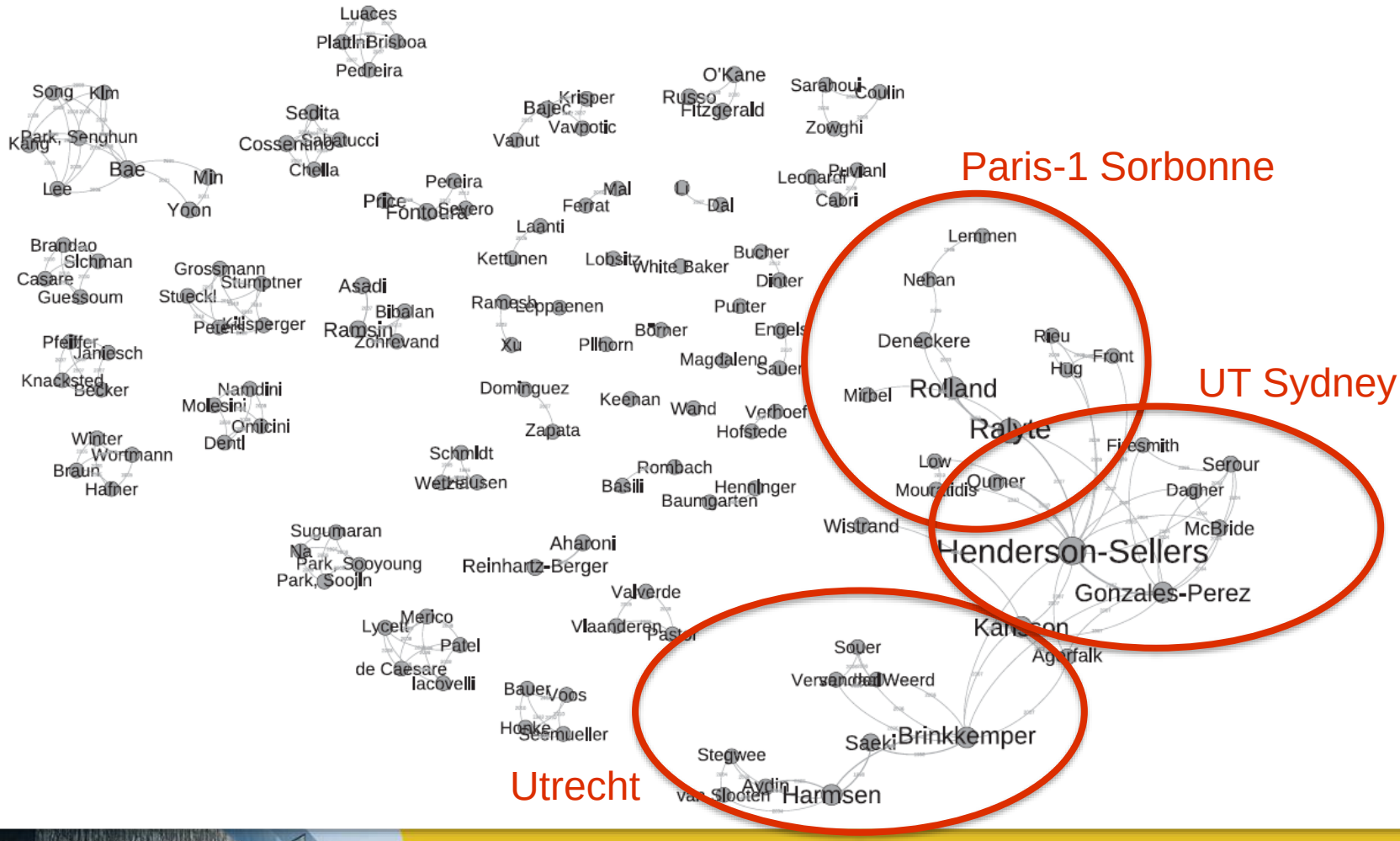
Go to Google Home - About Google - About Google Scholar

©2011 Google



"If I have seen further it is only by standing on the shoulders of giants." (Newton, 1667)

World-wide researcher network



A mapping study on the feasibility of method engineering
M Kuhrmann, D Méndez Fernández, M Tiessler - Journal of Software: Evolution and Process, 2014

Defining: Method engineering

Term coined by Kumar & Welke (GSU, Atlanta, 1992):

> "There is no detailed information systems methodology which is best in all situations." → solution: method engineering

Definition by Brinkkemper (Twente, 1996):

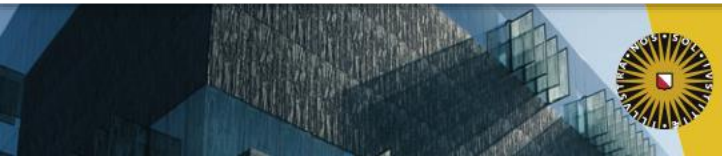
> "Method engineering is the engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems."

Situational method engineering

"How to adapt your method to a particular project situation?"

- Harmsen, Brinkkemper & Oei (1994): A situational method is "an information systems development method tuned to the situation of the project at hand"

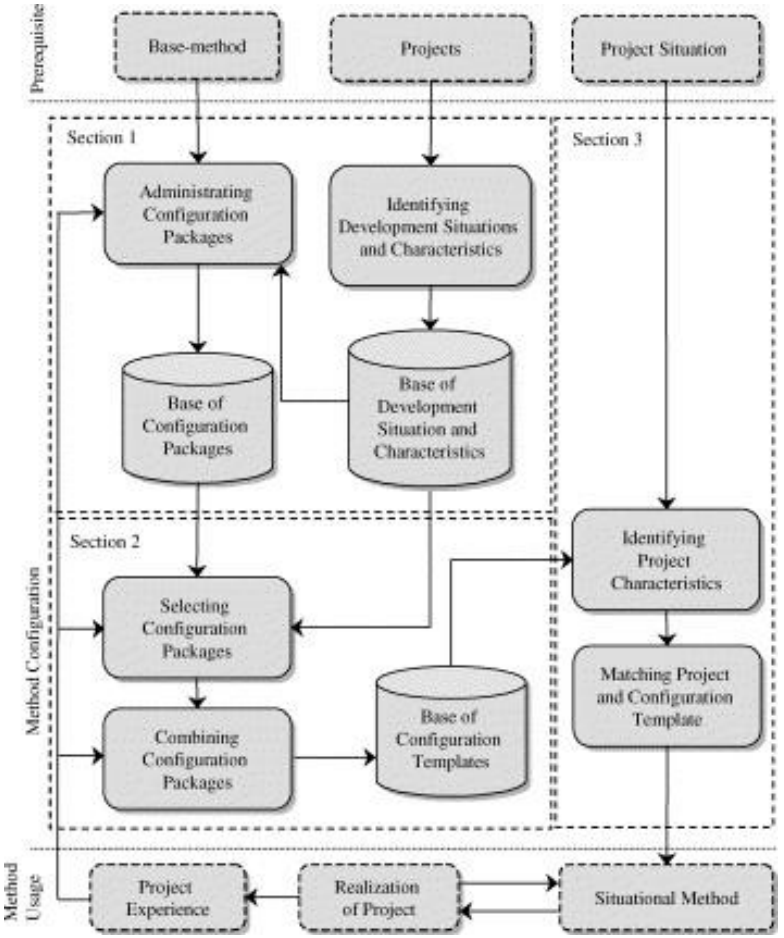
- Ralyté, Deneckère, Rolland (Sorbonne, 2003): adapt methods to the project situation at hand



Method configuration

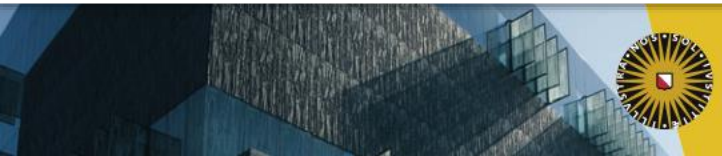
- To adapt a particular method to various situated factors.
- Focus is on one method which is configured in a particular situation, instead of using a set of methods as a base for assembly.

Karlsson (Uppsala, 2002); Karlsson and Ågerfalk (2004)



Incremental method engineering

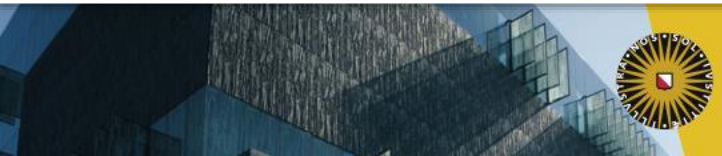
- "How do methods evolve over time, and how should this be supported?"
- Rossi, Ramesh, Lyytinen & Tolvanen (Jyvaskyla, 2004): "How to preserve method rationale?"
- van de Weerd, Brinkkemper & Versendaal (2010): "Which increments occure most frequently? What are the main drivers for method increments?"



Outline

- We are standing on the shoulders of giants…
  - Method engineering
  - Creation of methods
  - Tool support
  - Method comparison

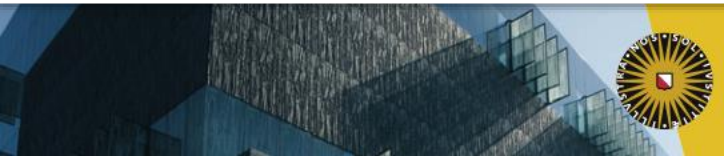
- …but do not always agree with each other



Why method construction?

- To develop a new method to satisfy a set of situational requirements
- To add alternative ways-of-working in a method
- To extend a method by a new functionality

(Ralyté & Rolland, 2003)



Harmsen, Brinkkemper & Oei (1994)

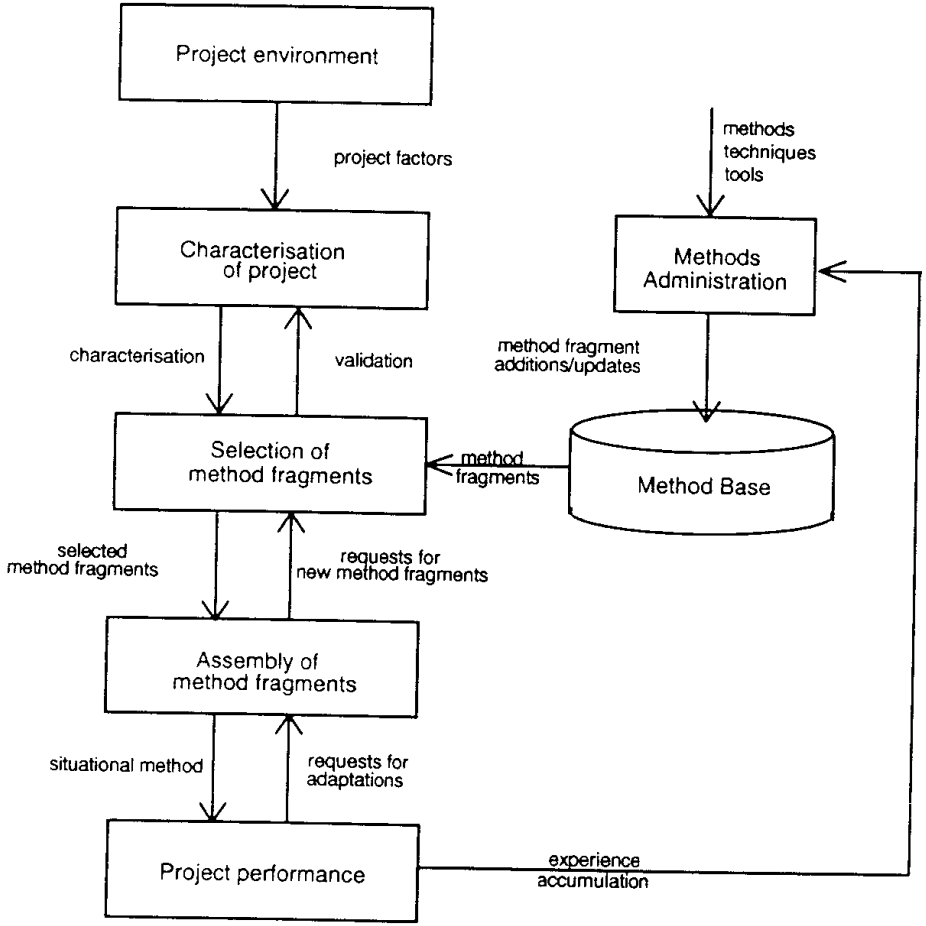
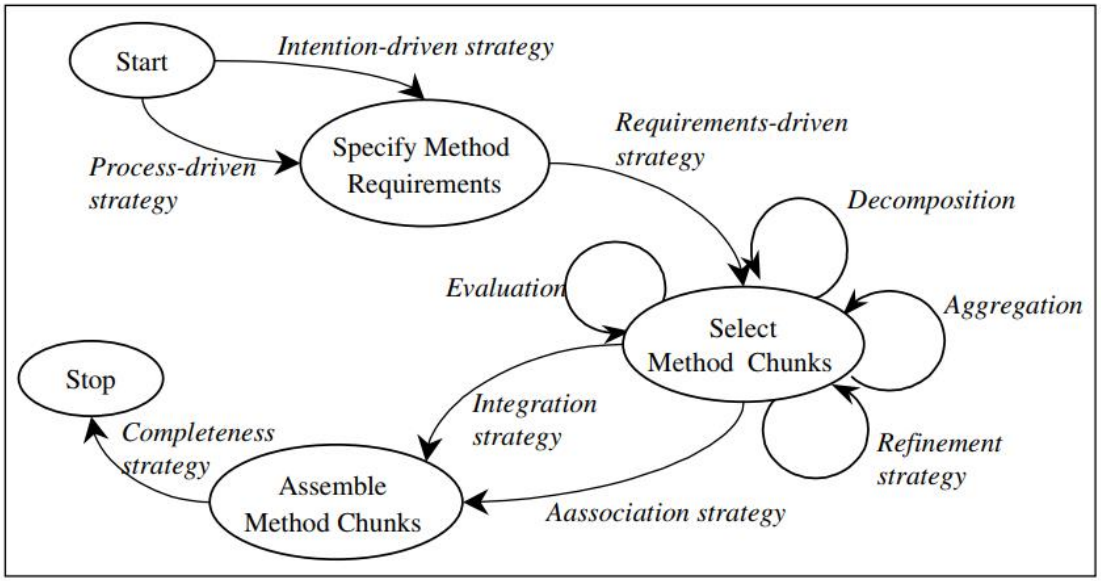


Figure 1 The process of configuration of situational methods

An assembly-based process model for situational method engineering

1. Specify method requirements
2. Select method fragments
3. Assemble method fragments

(Ralyté, Deneckère & Rolland, 2003)



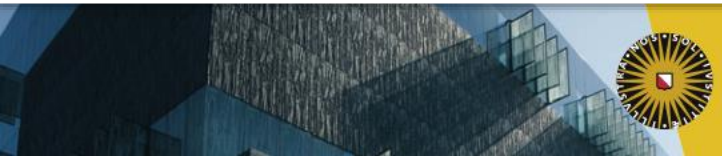
Adopted and extended by Van de Weerd et al. (2006)…

1. Analyze implementation situations and identify needs.

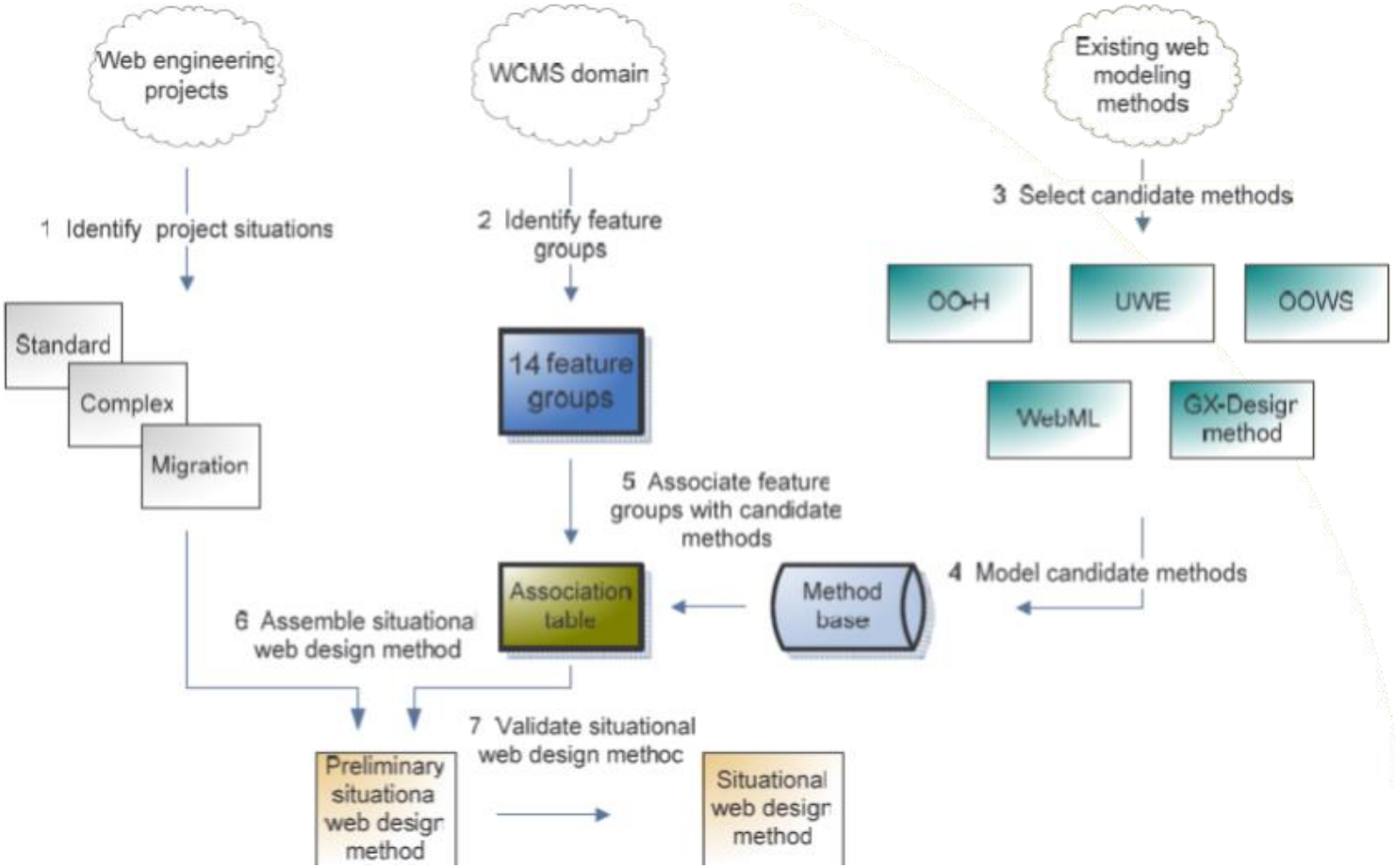
2. Select candidate methods that meet one or more aspects of the identified needs.

3. Analyze candidate methods and store relevant method fragments in a method base.

4. Assemble a new method from useful method fragments and use route map configuration to obtain situational methods



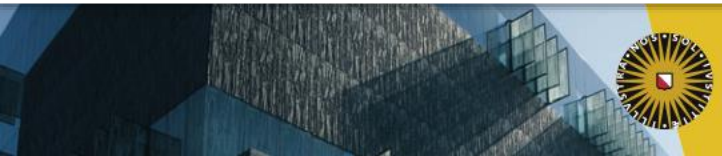
...adopted and extended by Luinenburg (2008)



Outline

- We are standing on the shoulders of giants…
  - Method engineering
  - Creation of methods
  - Tool support
  - Method comparison

- …but do not always agree with each other

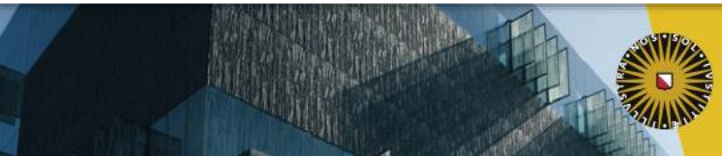


Tool support

- CASE: Computer Aided Software Engineering
- CAME: Computer Aided Method Engineering

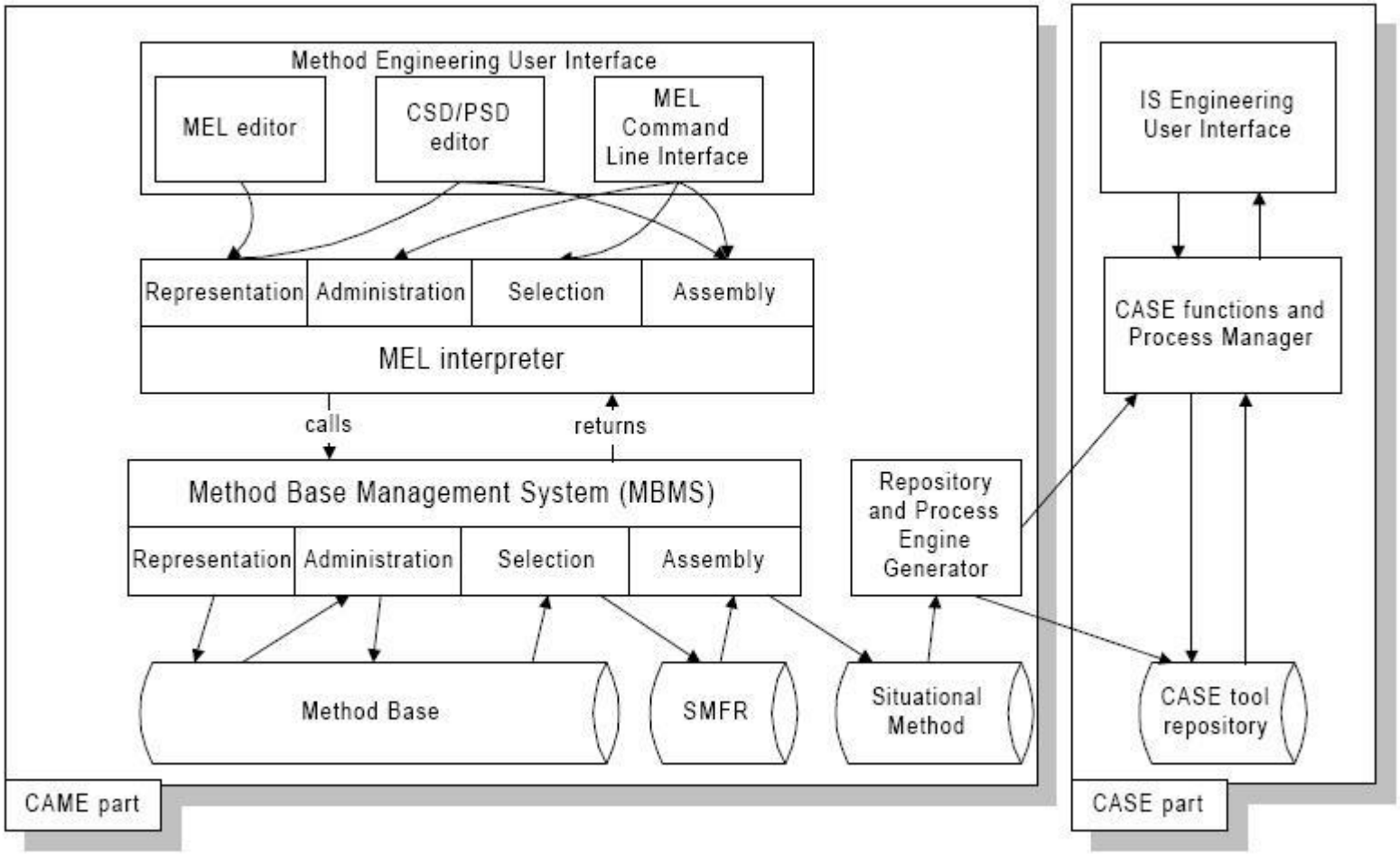
A CAME tool can be used for three purposes:
1) to specify methods to be used in a method environment,
2) to compare, analyze, and select methods; and
3) to store the accumulated knowledge of methods and situation factors.

Martiin, Harmsen & Rossi (1996)



Decamerone (Twente)

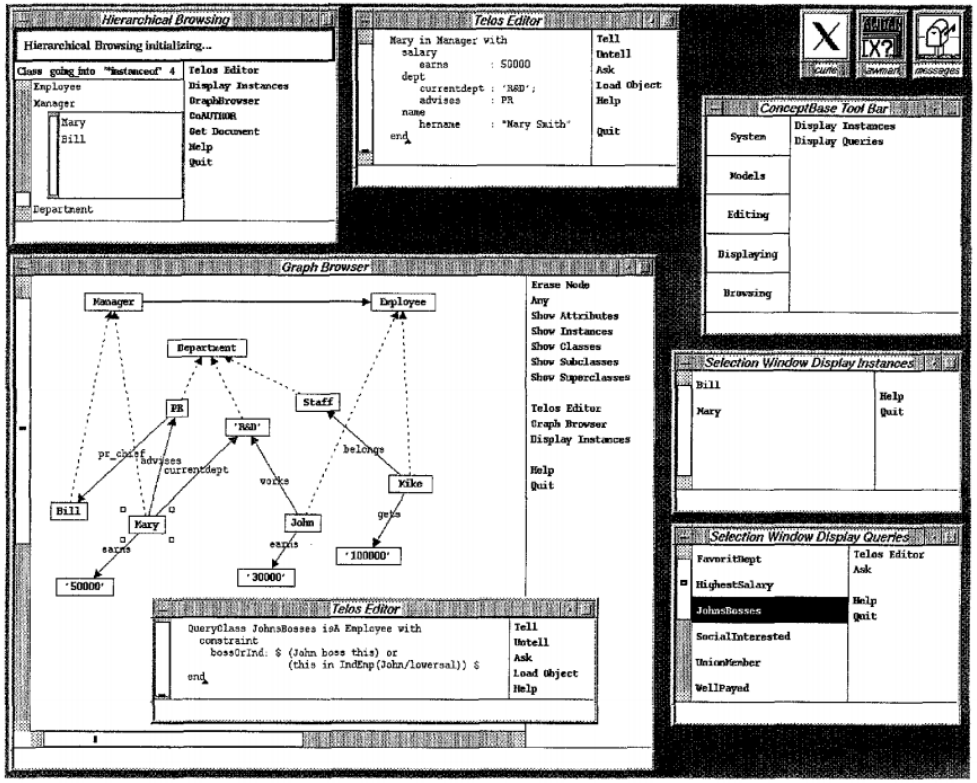
- Harmsen, Situational Method Engineering (dissertation,



ConceptBase (Aachen)

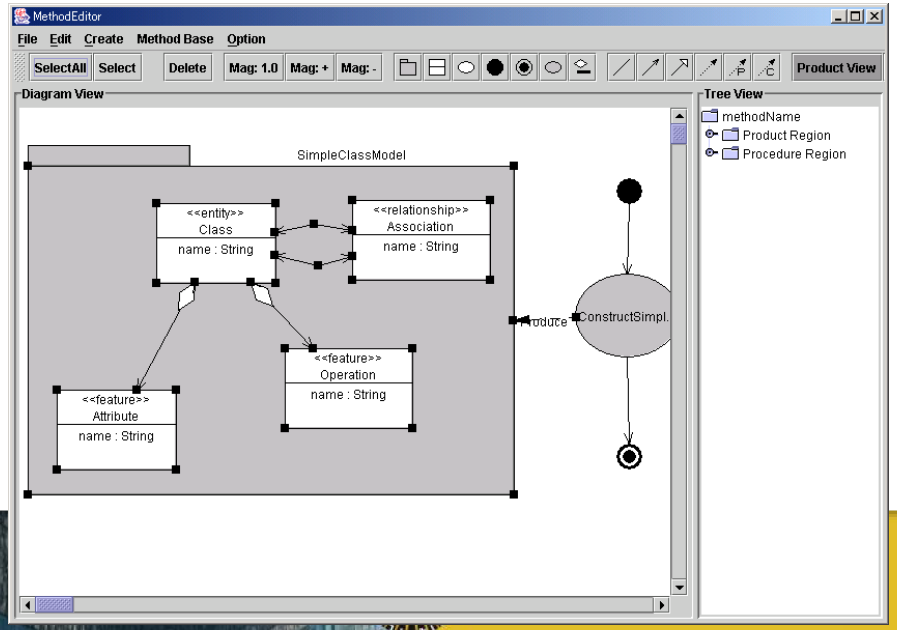
- Database system for meta modeling and method engineering.
- Based on an object-centered data model, that is manipulated by using logic (in the textual interface) or by drawing (in the graphical user interface).
- First-order logic formulas are used to express constraints and queries.

Jeusfeld, Jarke, Nissen & Staudt (1998)



MethodBase (Tokyo)

- Product and process fragments are represented with a simplified version of UML class diagrams and activity diagrams respectively.
  (→ predecessor of PDDs)
- In addition, constraints on method fragments can be defined using the Object Constraint Language (OCL).

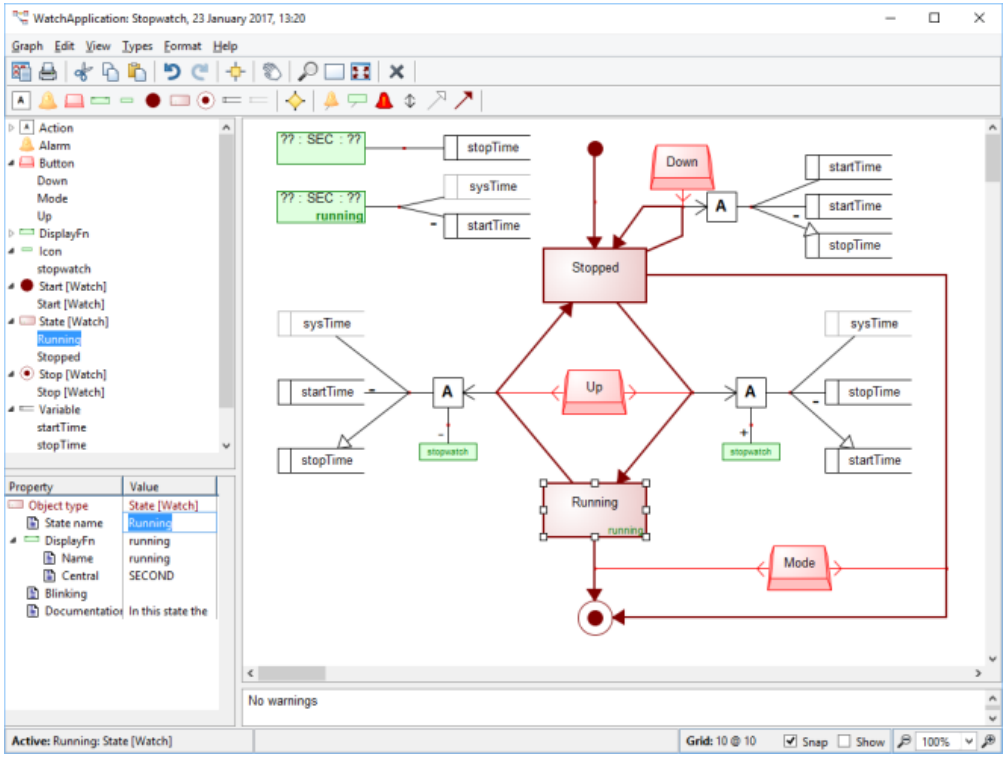


Saeki (2006)

MetaEdit+ (Jyvaskyla, Fi)

- Commercial tool
  - Workbench to create modeling technique
  - Modeling environment to work with the actual method

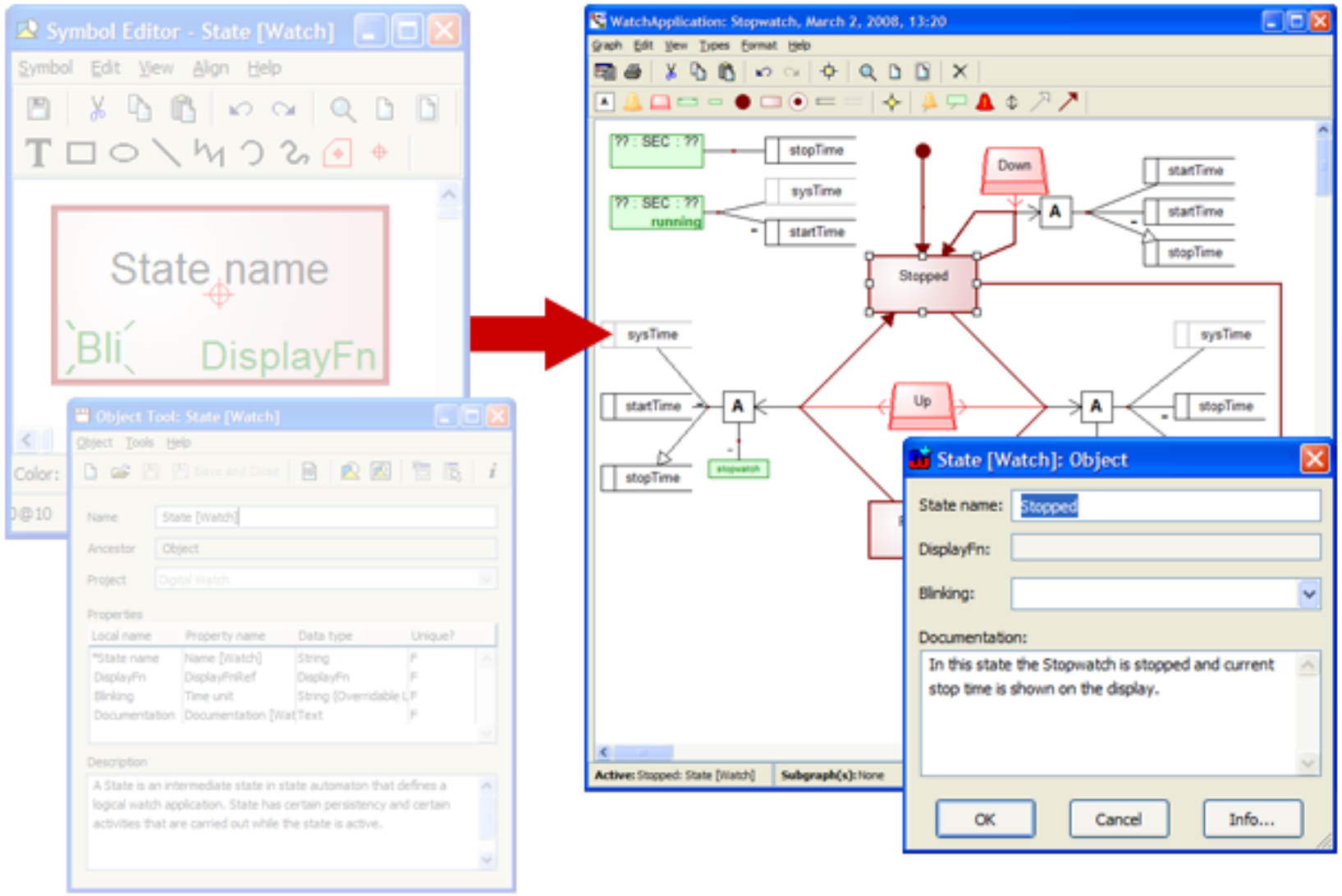
Kelly, Lyytinen & Rossi (1996)

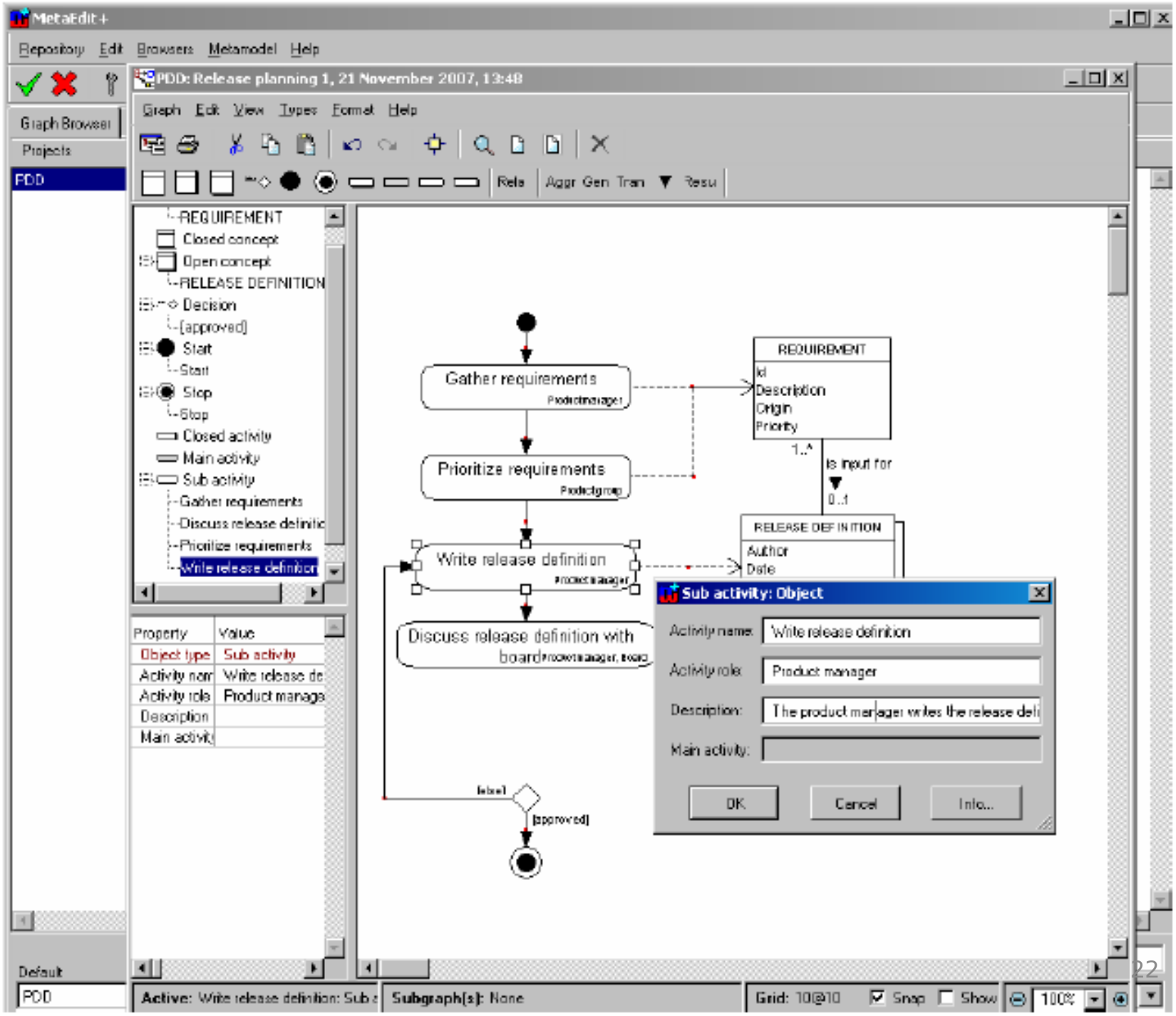


MetaEdit+ (2)

Design your modeling language

Use your modeling language

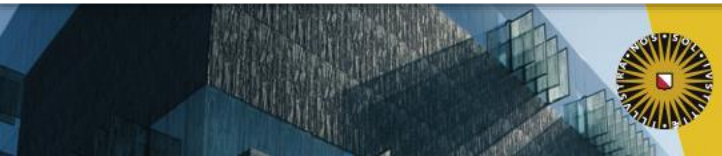



MetaEdit+
Repository Edit Browsers Metamodel Help
PDD: Release planning 1, 21 November 2007, 13:48
Graph Edit View Types Format Help
Graph Browser
Projects
PDD
REQUIREMENT
Closed concept
Open concept
RELEASE DEFINITION
Decision
[approved]
Start
Start
Stop
Stop
Closed activity
Main activity
Sub activity
Gather requirements
Discuss release definitic
Prioritize requirements
Write release definitior
Property
Value
Object type
Sub activity
Activity nam
Write release de
Activity role
Product manage
Description
Main activity
Gather requirements
Prioritize requirements
Write release definition
Discuss release definition with board
label
[approved]
REQUIREMENT
Id
Description
Origin
Priority
1..*
is input for
0..1
RELEASE DEFINITION
Author
Date
Sub activity: Object
Activity name: Write release definition
Activity role: Product manager
Description: The product manager writes the release def
Main activity:
OK
Cancel
Info...
Default
PDD
Active: Write release definition: Sub
Subgraph(s): None
Grid: 10@10
Snap
Show
100%

Outline

- We are standing on the shoulders of giants…
  - Method engineering
  - Creation of methods
  - Tool support
  - Method comparison

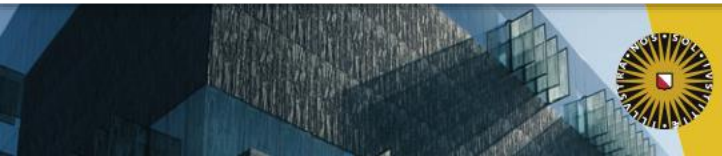
- …but do not always agree with each other



But we do not always agree…

- Method fragments vs. method chunks
  - Method fragment (Brinkkemper, Saeki, Harmsen, Weerd, etc.)
  - Method chunks (Ralyté, Henderson-Sellers, Rolland, Jeusfeld, etc.)
  - Method components / OPF (Henderson-Sellers)

- Model vs. meta-model

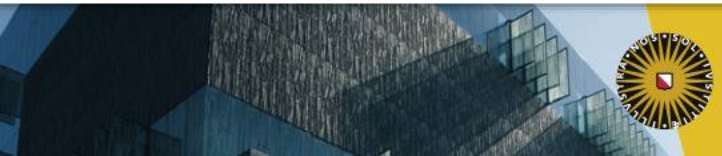


Current trends

- MaaS: Method as a service (Rolland et al., 2007)
  Using Web service technology to provide self-describing, platform agnostic elements (MaaS), accessible through standard interfaces and that can be assembled together

- Situationality (Bekker et al., 2008, 2009, 2010)
  Identifying and using situational factors that influence the method fragment selection process

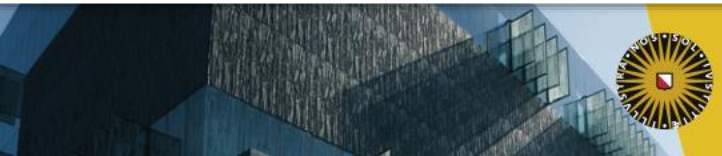
- Incremental method (Weerd et al., 2007, Vlaanderen et al., 2012)
  Creating small changes of a method to support gradual improvement



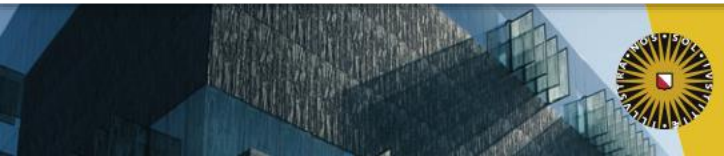
Current research

- Integration of DevOps with Scrum (Centric)
  - Capability identification in DevOps
  - Maturity benchmark in large scale software production
- Reference method for Digital Strategy (AndersonMacGyver)
  - Formalizing existing approach
  - Providing scientific justifications

- Open projects for master's thesis research
  - Methods and techniques for software architecture and requirements engineering
  - User stories
  - Continuous architecting
  - Artefacts in agile methods



QUESTIONS?



References

Jeusfeld, M.A., M. Jarke, & Mylopoulos, J. (2009*). Metamodeling for Method Engineering.* Camebridge, USA: The MIT Press.

Karlsson, F., & Ågerfalk, P. J. (2004). Method configuration: adapting to situational characteristics while creating reusable assets. *Information and Software Technology*, 46(9), 619-633.

Hong, S., Goor, G. van den, Brinkkemper, S. (1993). A formal approach to the comparison of object-oriented analysis and design methodologies. Proceedings of the 26th Annual Hawaii International Conference on System Sciences, Hawaii, 689–699.

Ralyté, J., Deneckère, R., & Rolland, C. (2003). Towards a generic model for situational method engineering. *Lecture Notes in Computer Science 2681*, 95-110.

Kevin Vlaanderen, Sjaak Brinkkemper, Inge van de Weerd: On the Design of a Knowledge Management System for Incremental Process Improvement for Software Product Management. *Int J Information Systems Modeling and Design* 3(4): 46-66 (2012)