

Method Engineering : Towards Methods as Services



Outline

- ❖ Backdrop
- ❖ MaaS

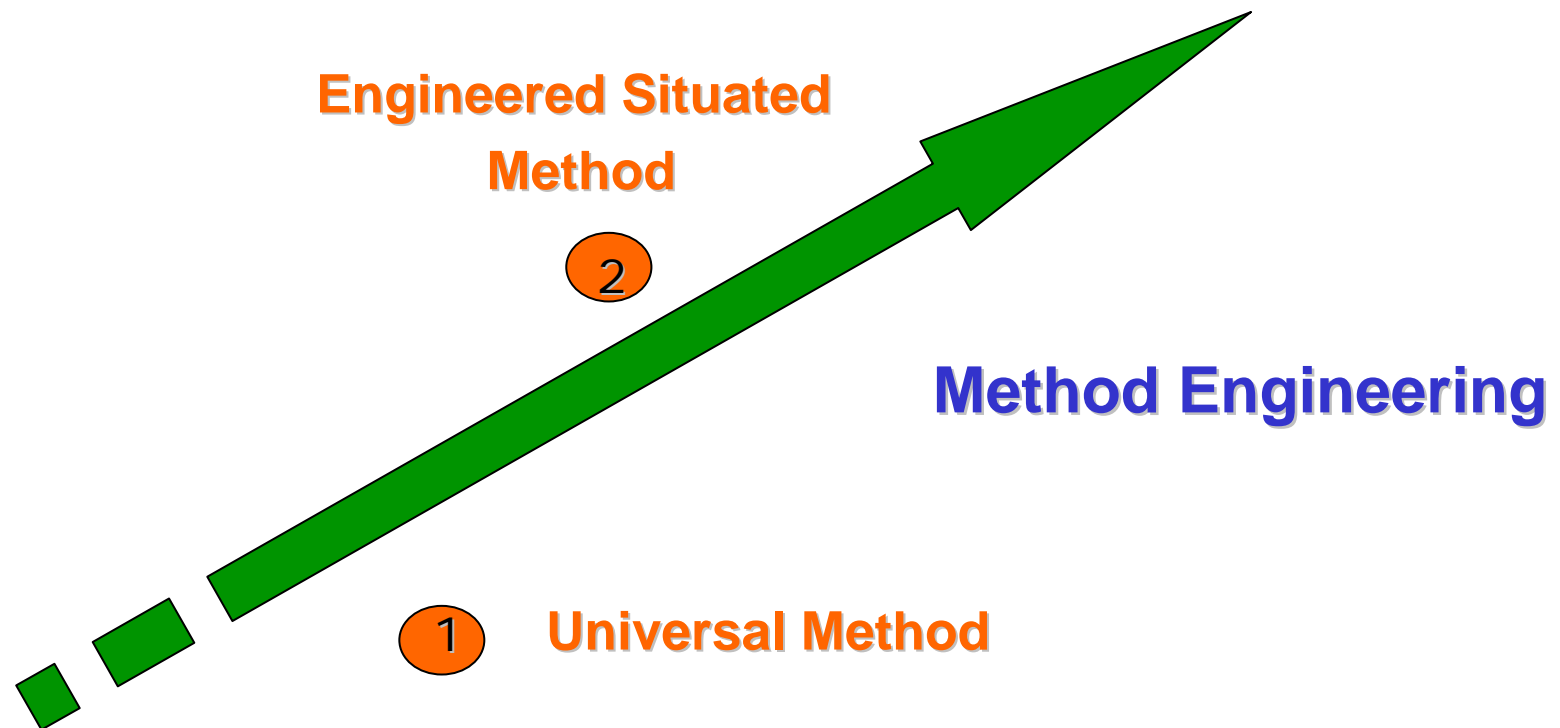
Prof. Colette Rolland

CRI

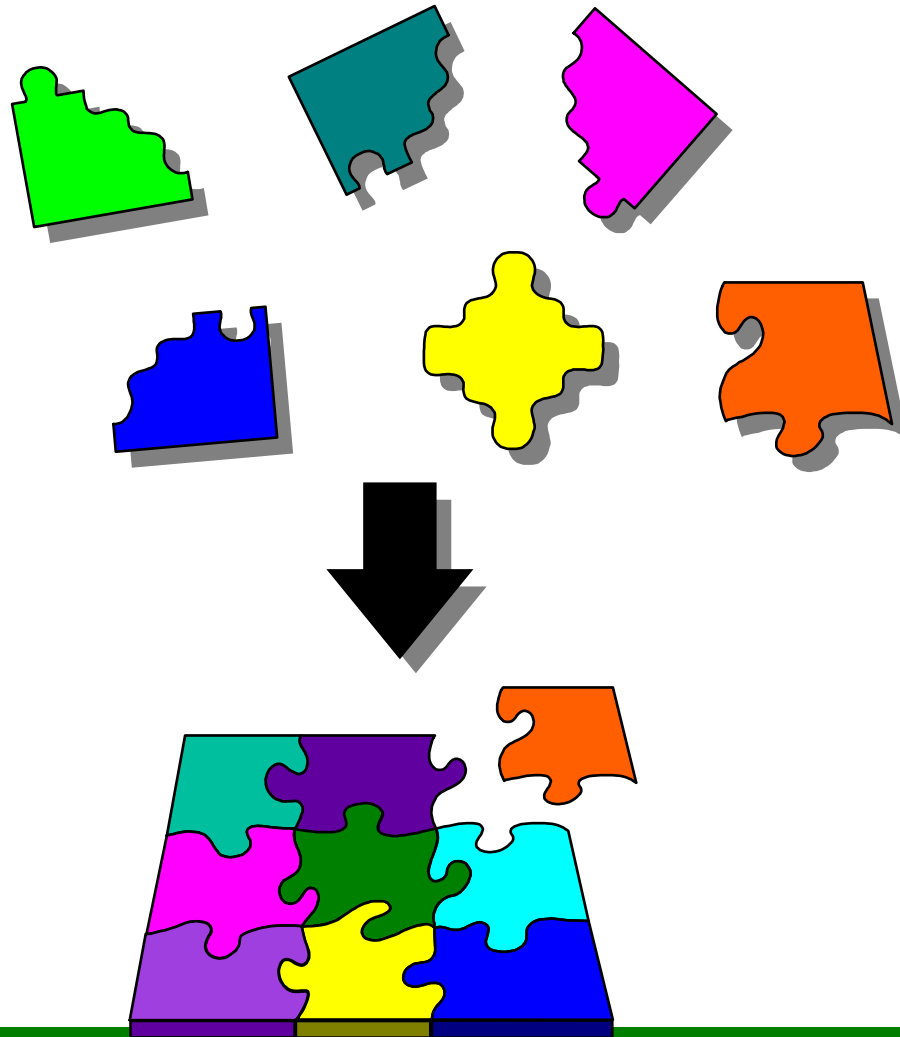
Université Paris1 Panthéon Sorbonne



Evolution in Information System Development Methods (ISD)



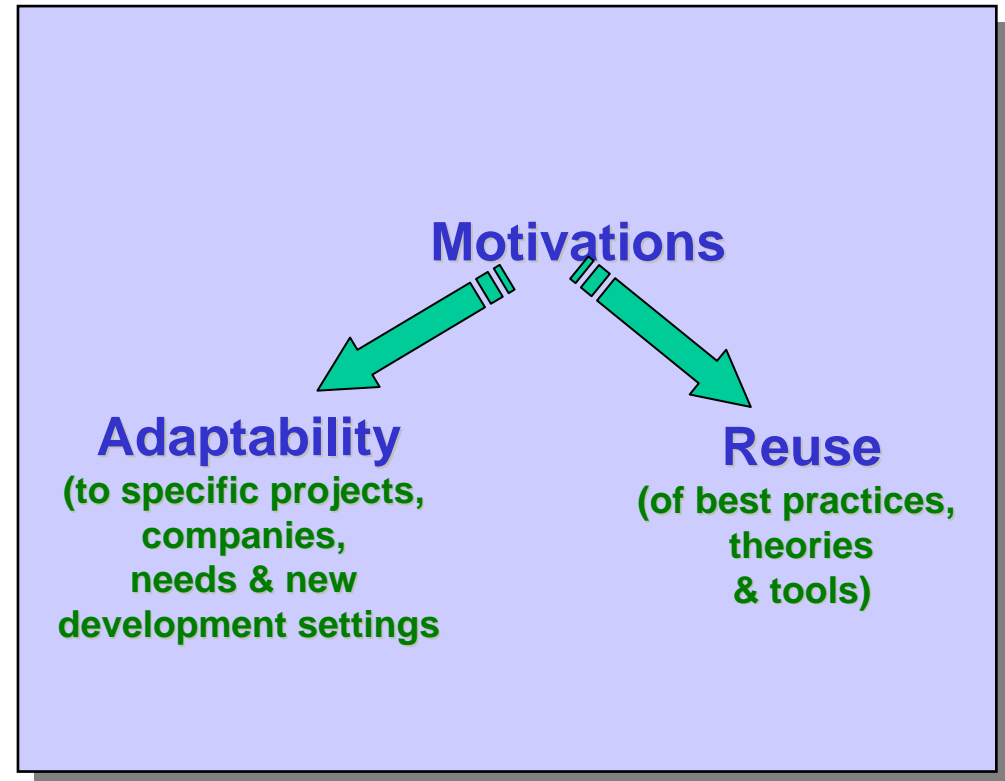
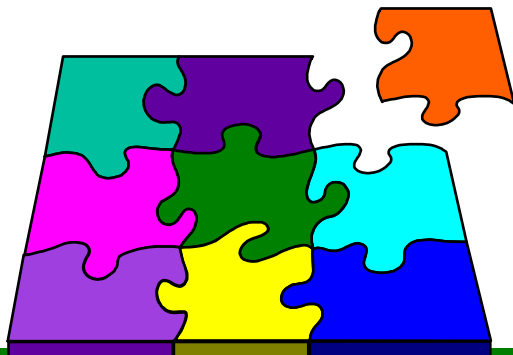
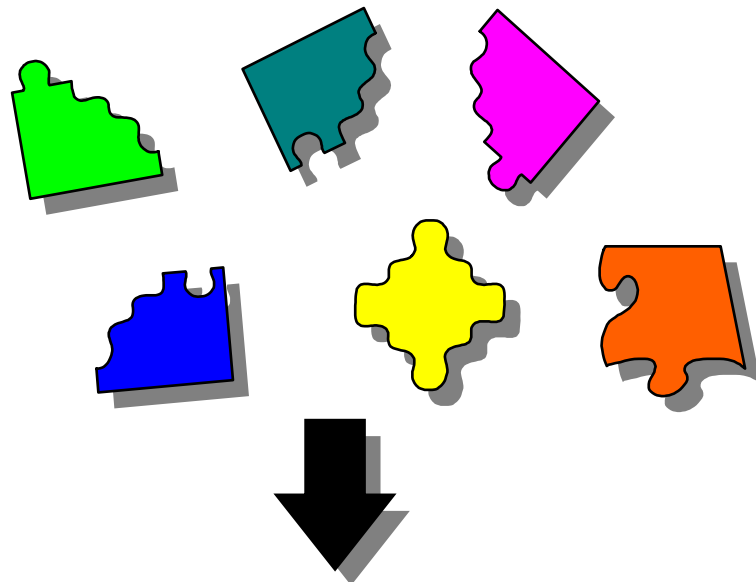
Method Engineering



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

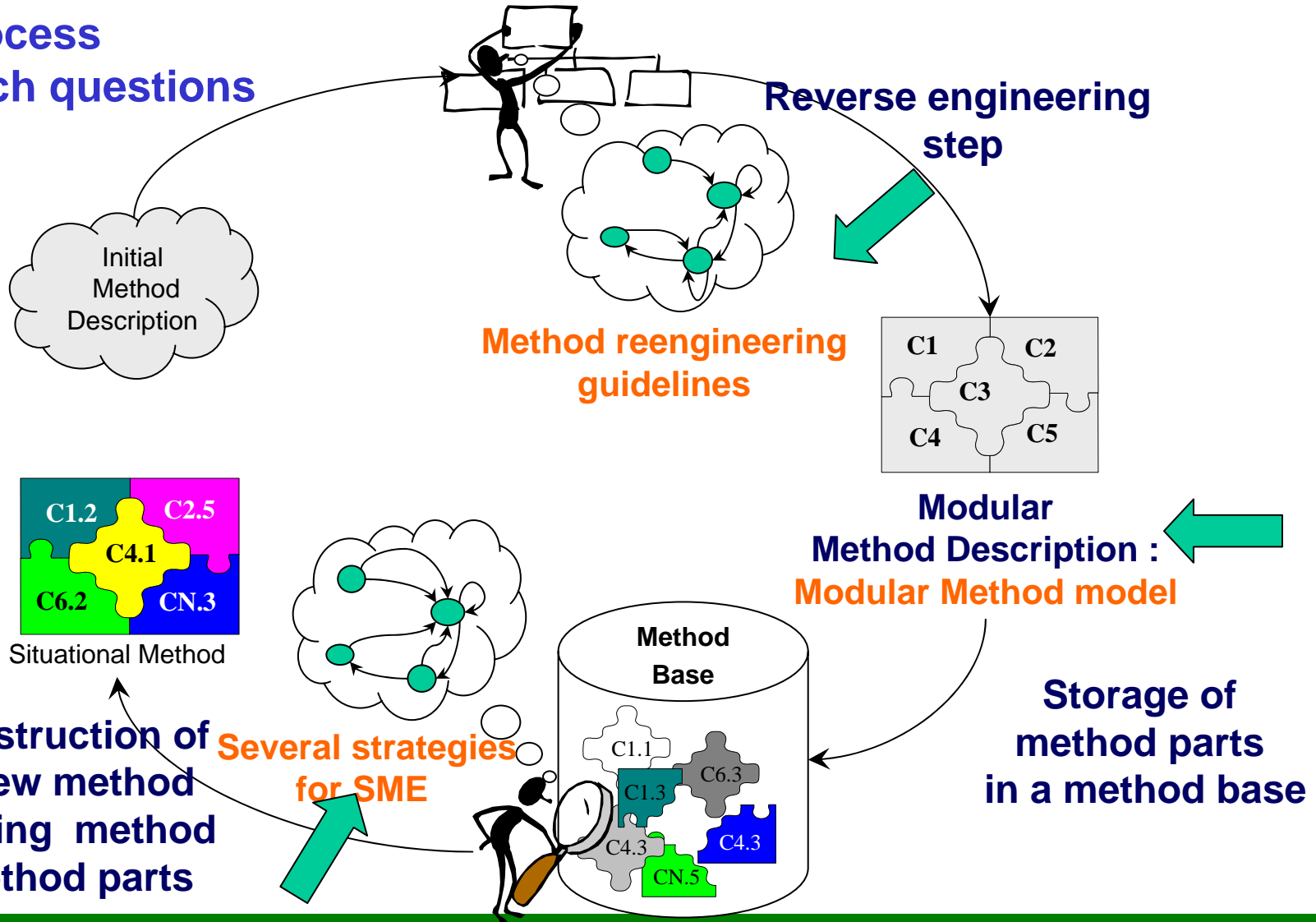
[Brinkkemper 96, Leppanen 2006]

Method Engineering



Method Engineering

♣ ME process
& research questions



Method Engineering

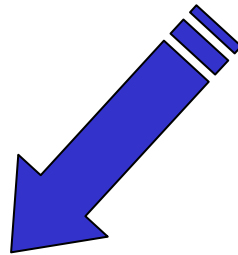
♣ Software engineering vs Method Engineering *

Software engineering	Method engineering
Modelling (specification)	Meta- Modelling
CASE (Computer Aided S/W Engineering)	CAME (Computer Aided Method Engineering)
Software Base	Method Base
Software metrics	Method metrics

*Motoshi Saeki talk (EMISE)

Method Engineering

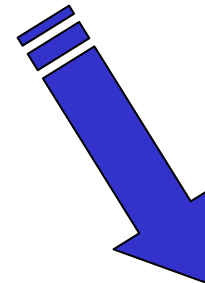
♣ Key Method Engineering Artefacts*



ME products

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

Harmsen97, Heym& Osterle,92, venable93, Kelly et al 96,
Graham et al 97,
OMG 05, Prakash 99, Ralyte 01,
* Leppanen06



ME processes

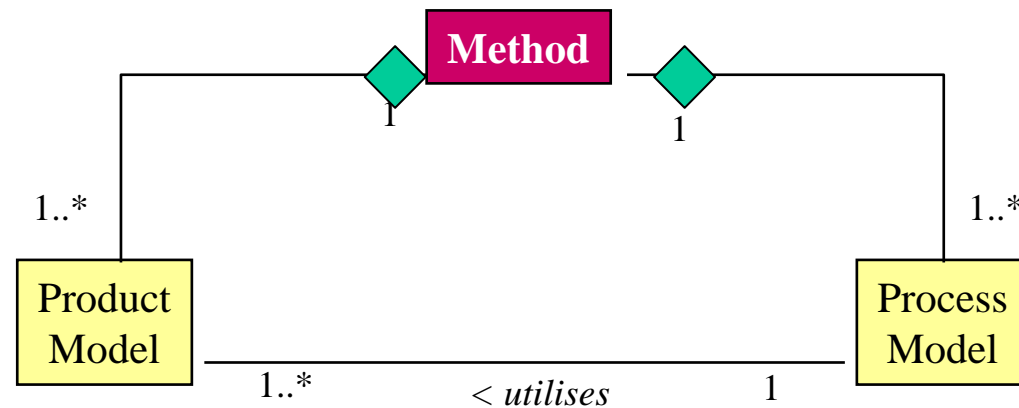
**ME strategies, approaches,
workflows to combine, integrate,
assemble method parts & to guide
the ME process**

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

Method Engineering Products

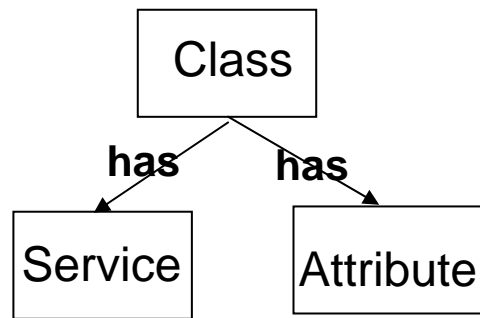
A modularization issue based on two assumptions

- ❖ (1) A method is composed of a product model and a process model



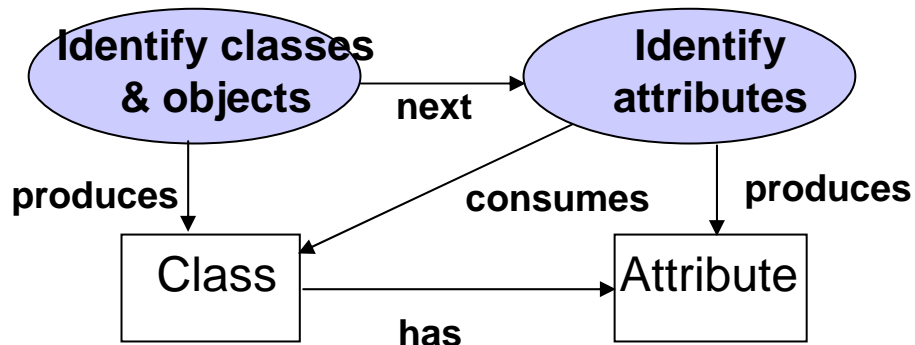
Method Engineering Products

- ♣ (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}

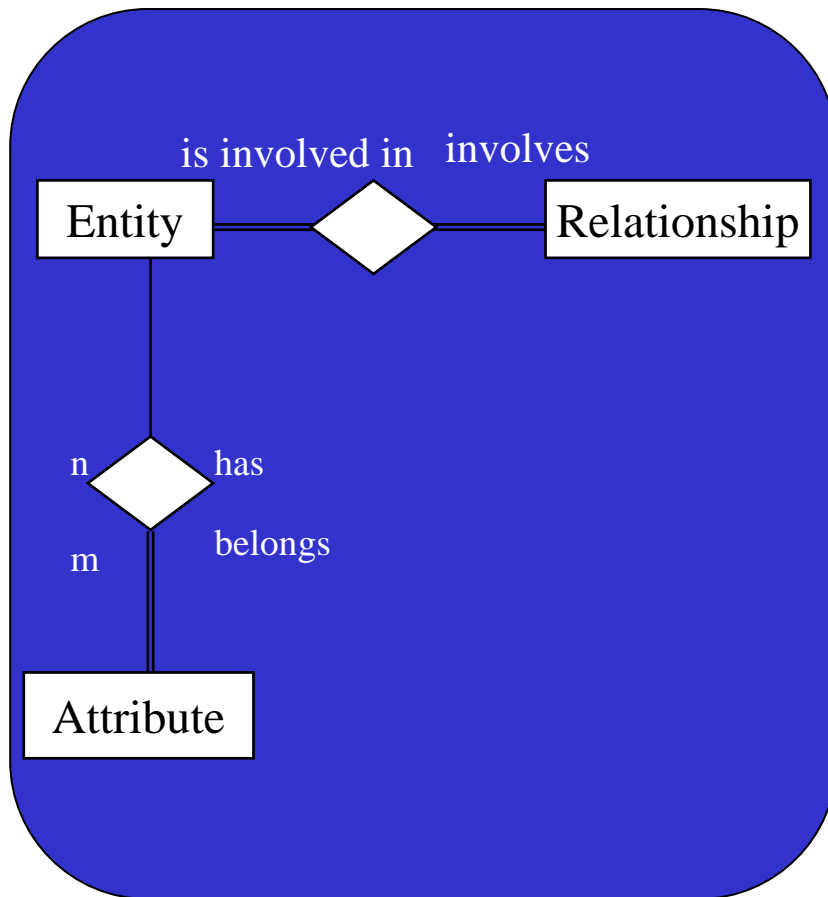
(b) Meta model of process part

*Motoshi Saeki talk (EMISE)

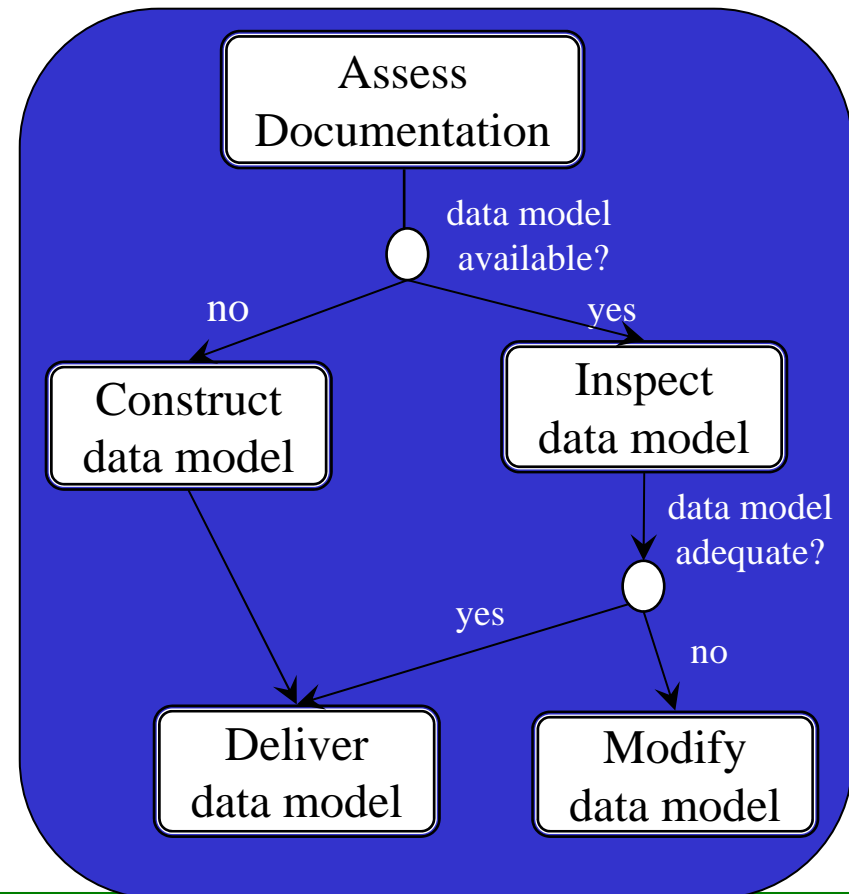
The Method Fragment Perspective

[Harmsen 94, Harmsen 97, Brinkkemper 99]

Product Fragment



Process Fragment

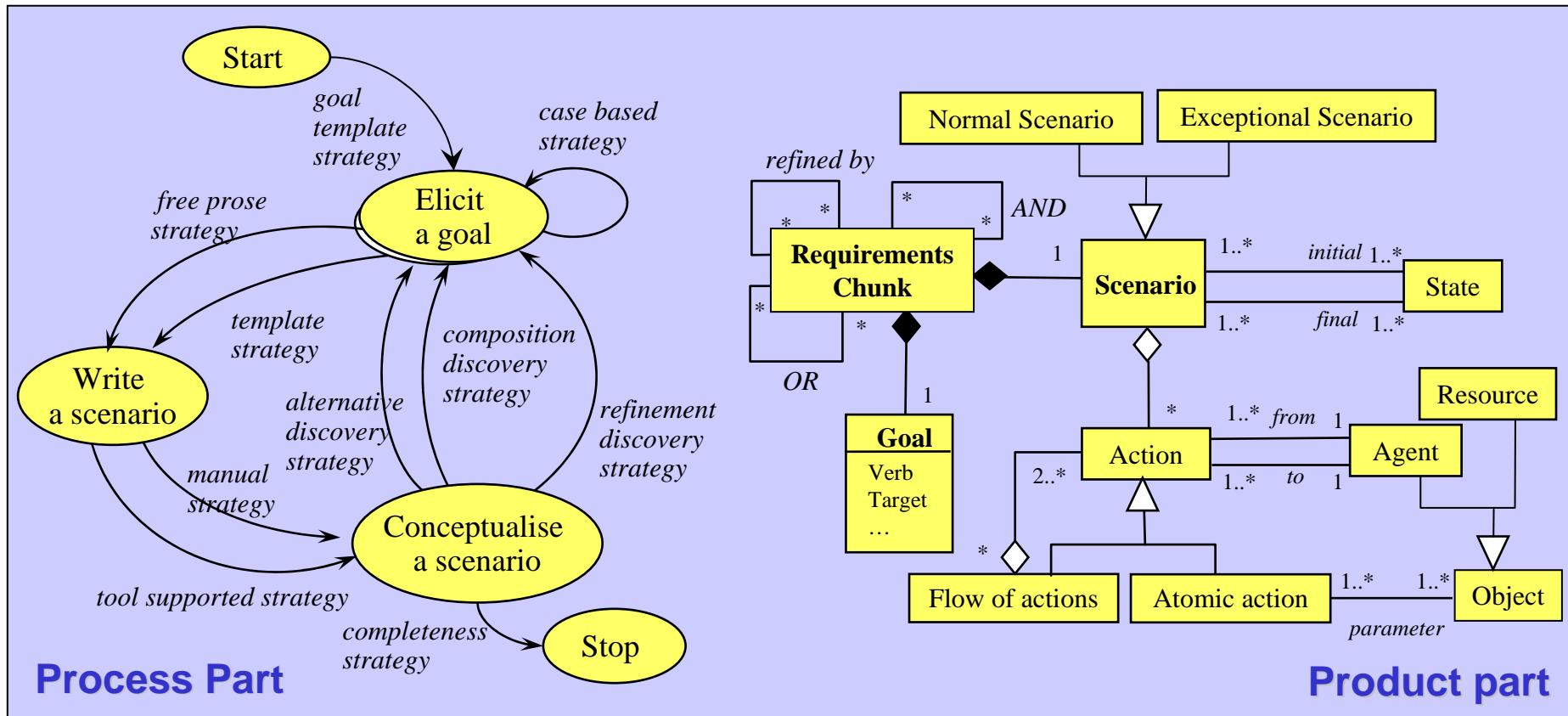


The Method Chunk Perspective

[Rolland & Prakash 1996, Rolland 1994, Ralyte & Rolland 2001]

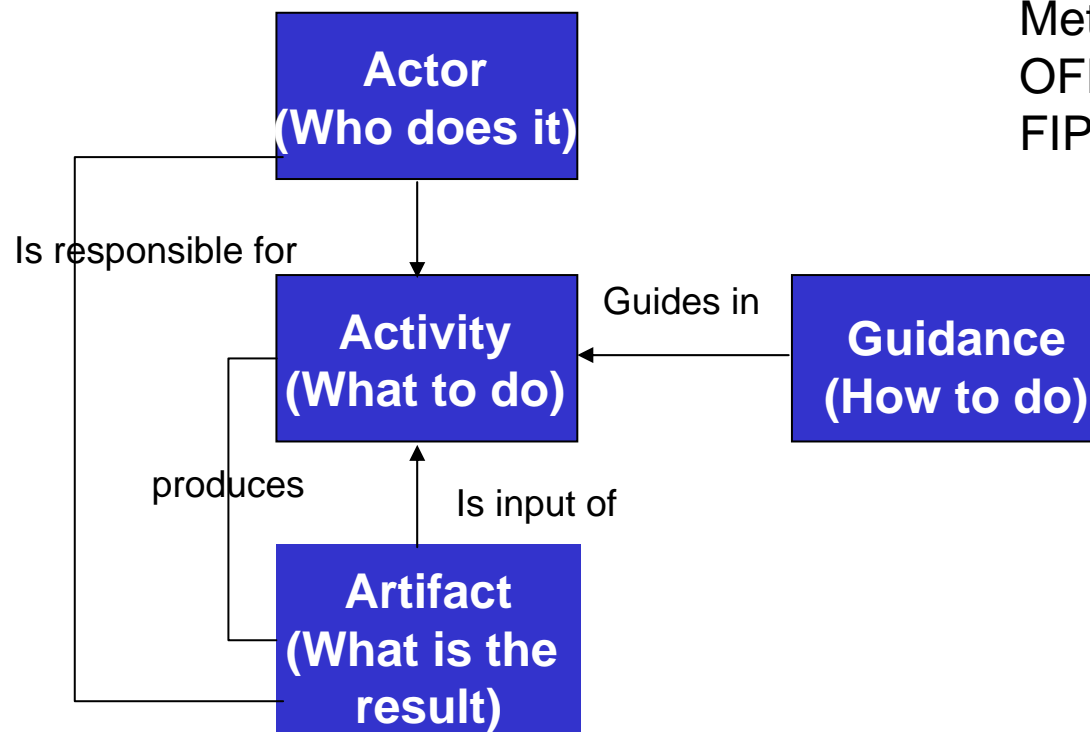
♣ **tight coupling in a chunk of the process and related product parts**

<(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

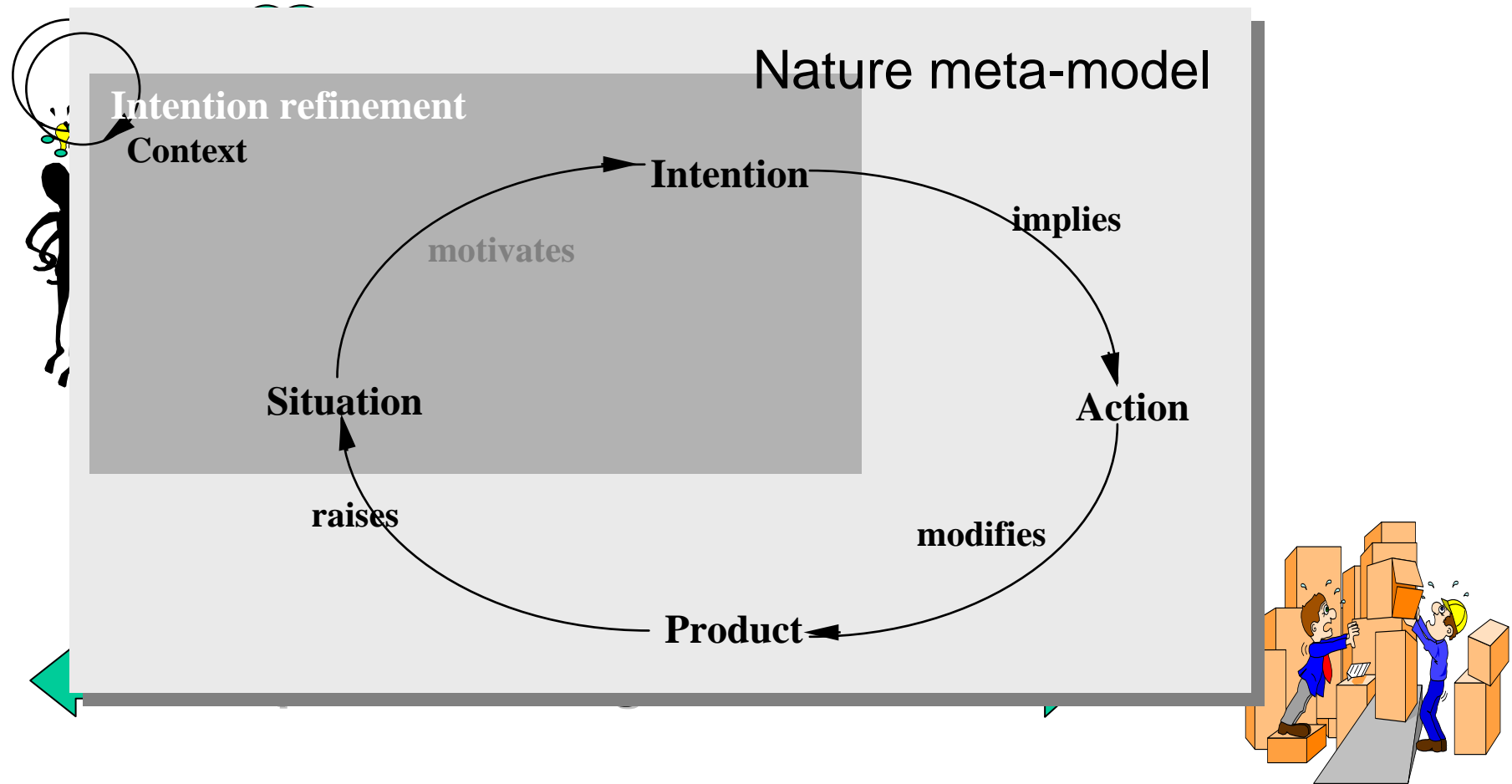
- **Intentionality is at the core of cognition** (all modern philosophers : Brentano, Twarsdwowski, Hurssel, Sartre, Merleau Ponty, Berner....)
- **Intention is a mental state that integrates desires and beliefs and determines actions**



Intentionality Model to Support Process Guidance

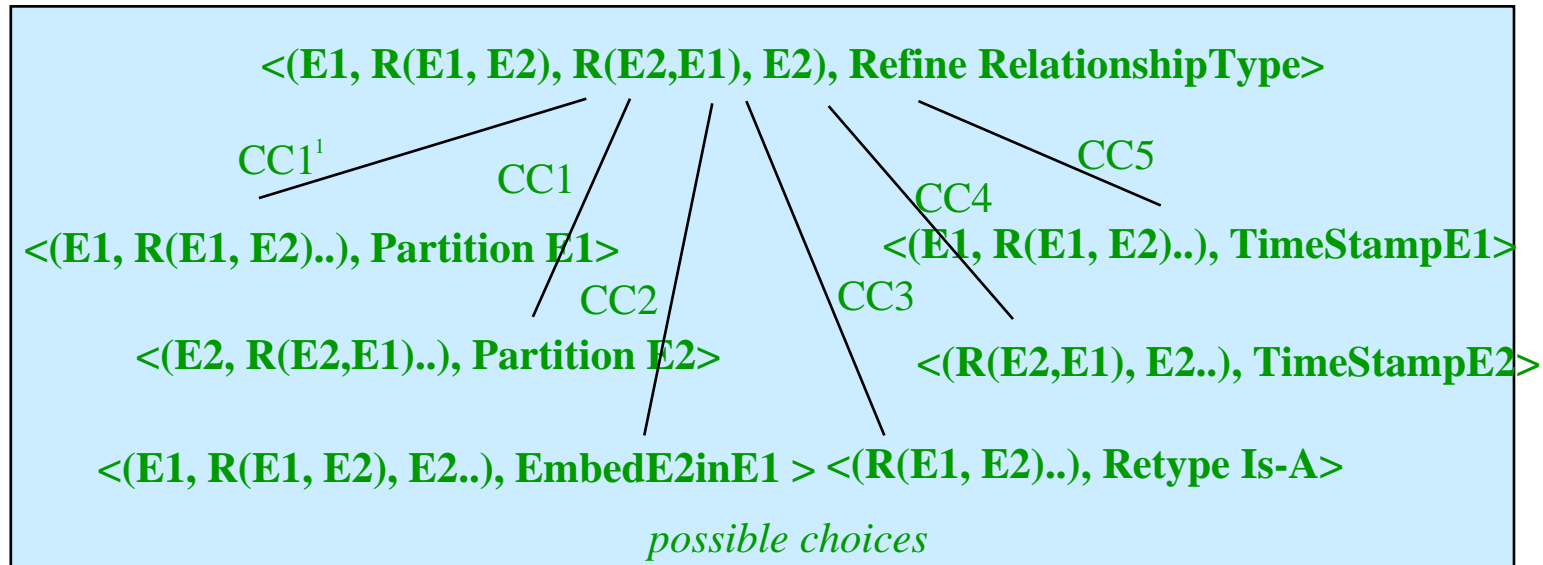
Method Engineering Products

♣ Guiding the achievement of intentions



Method Engineering Products

♣ Choice Guideline in NATURE



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: $\langle t, s, p - p, s, v \rangle$

a2: cardinalities are: $\langle p, ?, ? - ?, ?, ? \rangle$

a3: cardinalities are: $\langle ?, ?, ? - p, ?, ? \rangle$

a4: cardinalities are: $\langle t, m, p - t, s, p \rangle$

a5: cardinalities are: $\langle ?, ?, v - ?, ?, p \rangle$

a6: cardinalities are: $\langle ?, ?, p - ?, ?, v \rangle$

a7: cardinalities are: $\langle t, s, p - p, s, p \rangle$

a8: cardinalities are: $\langle p, m, v - p, m, v \rangle$

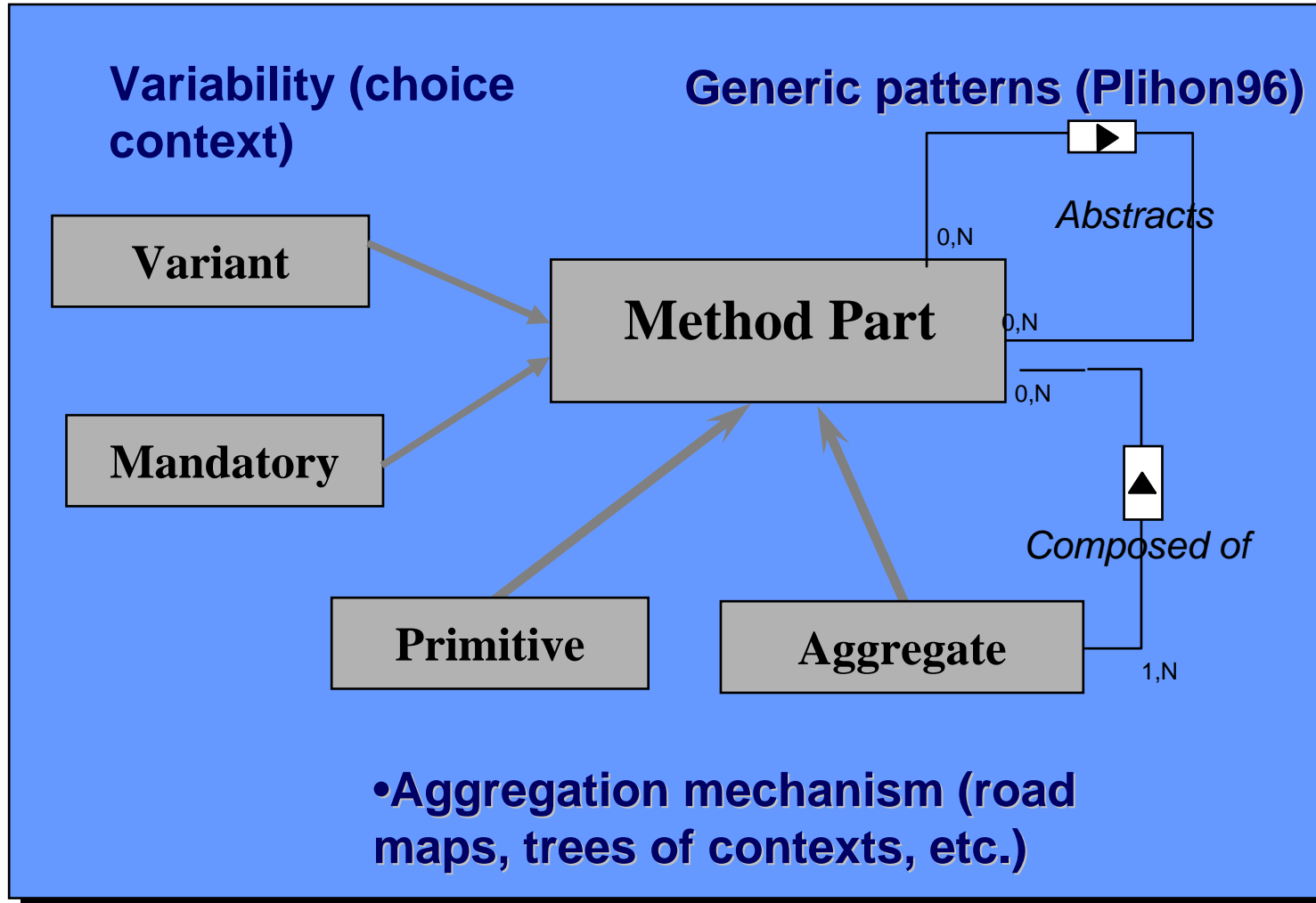
a9: cardinalities are: $\langle t, m, v - t, s, p \rangle$

a10: cardinalities are: $\langle t, s, p - t, s, p \rangle$

a11: cardinalities are: $\langle t, s, v - t, s, p \rangle$

Method Engineering Products

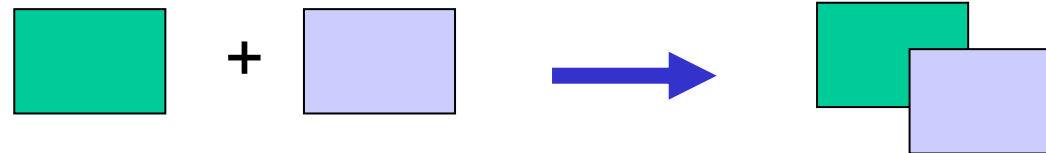
- ♣ Less consensual view on method part relationship types



Method Engineering Processes

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

□ Assembly based



□ Extension based



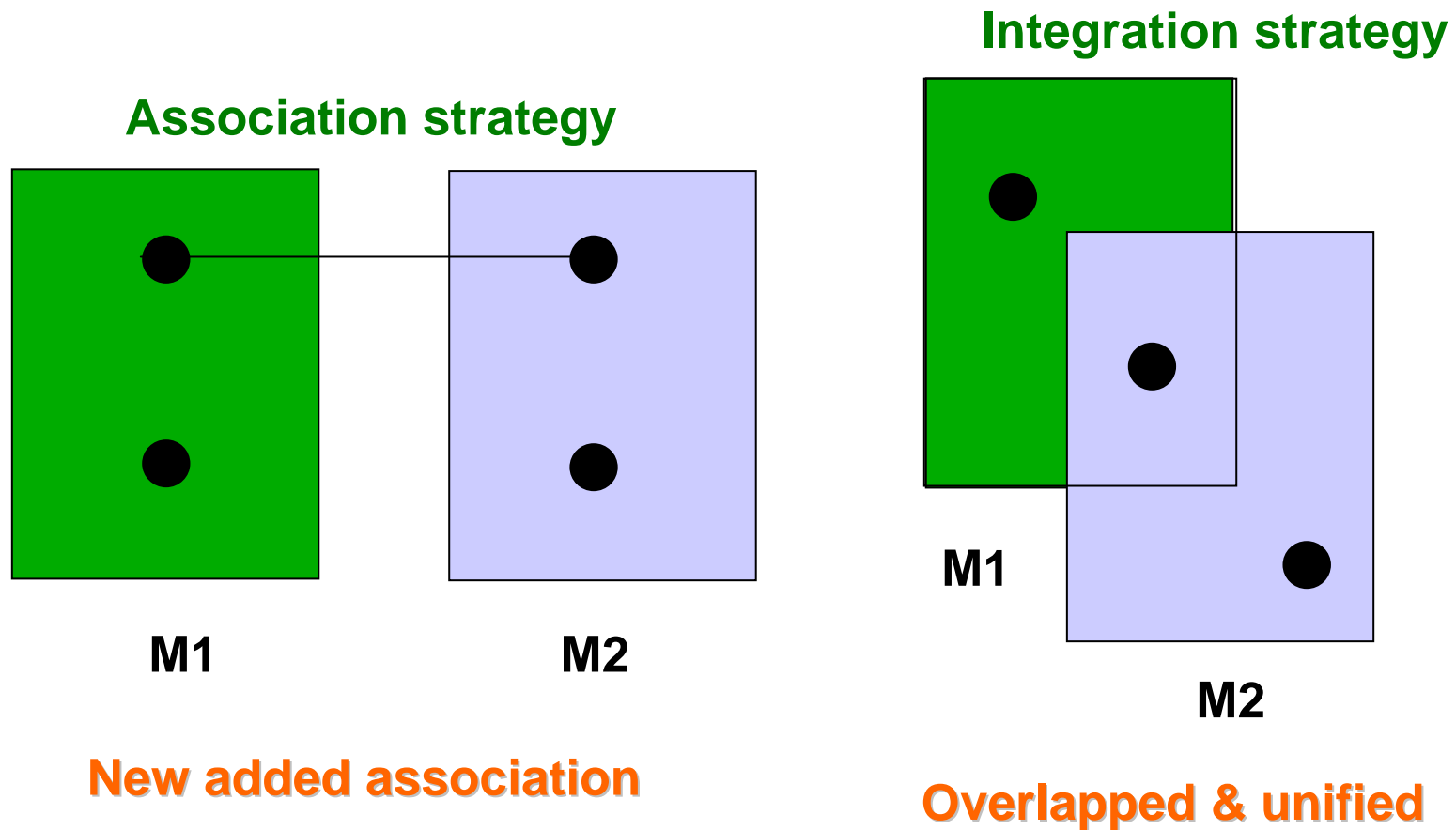
□ Paradigm based



*Motoshi Saeki talk in EMISE

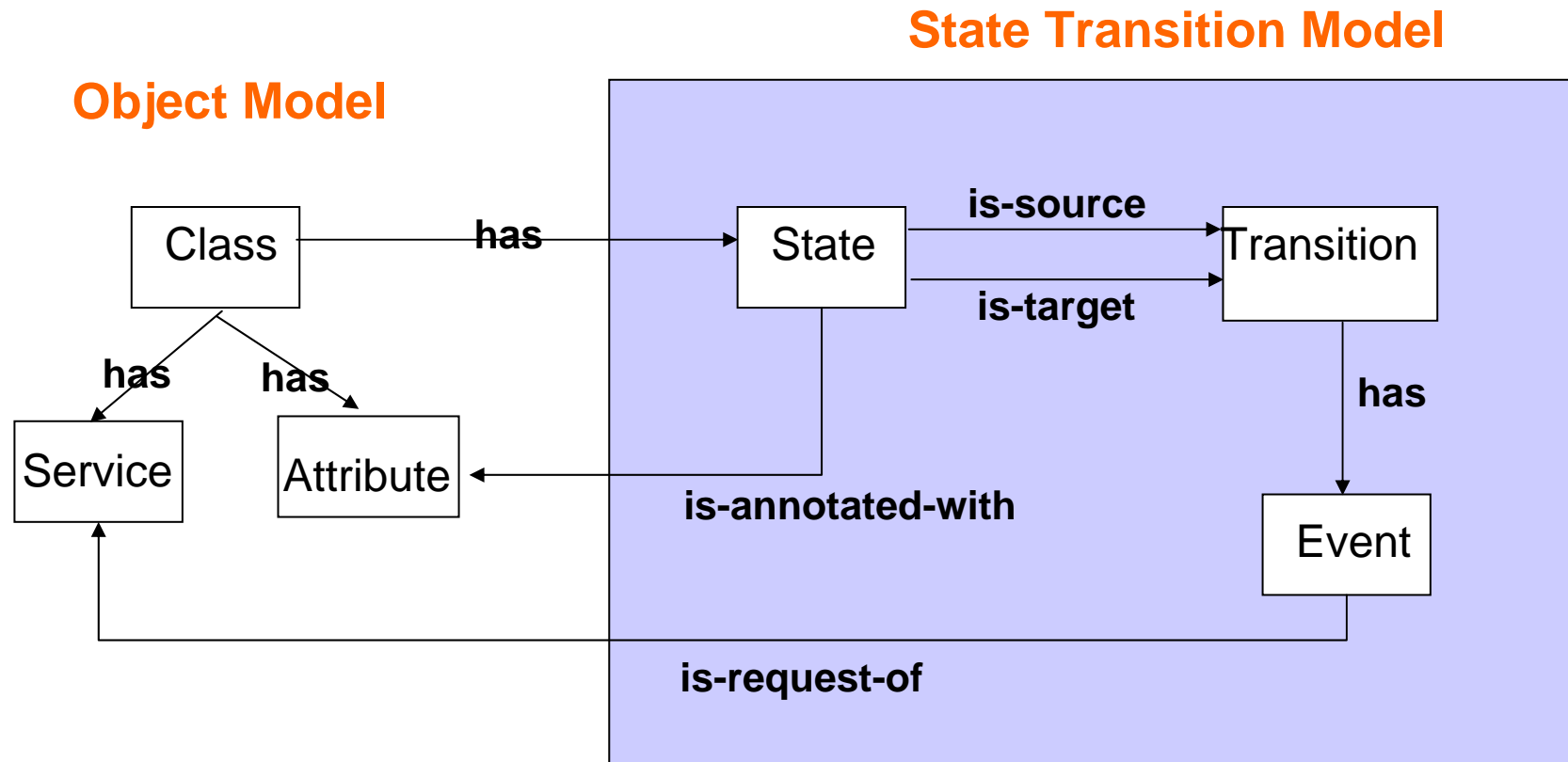
Method Engineering Processes

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



Method Engineering Processes

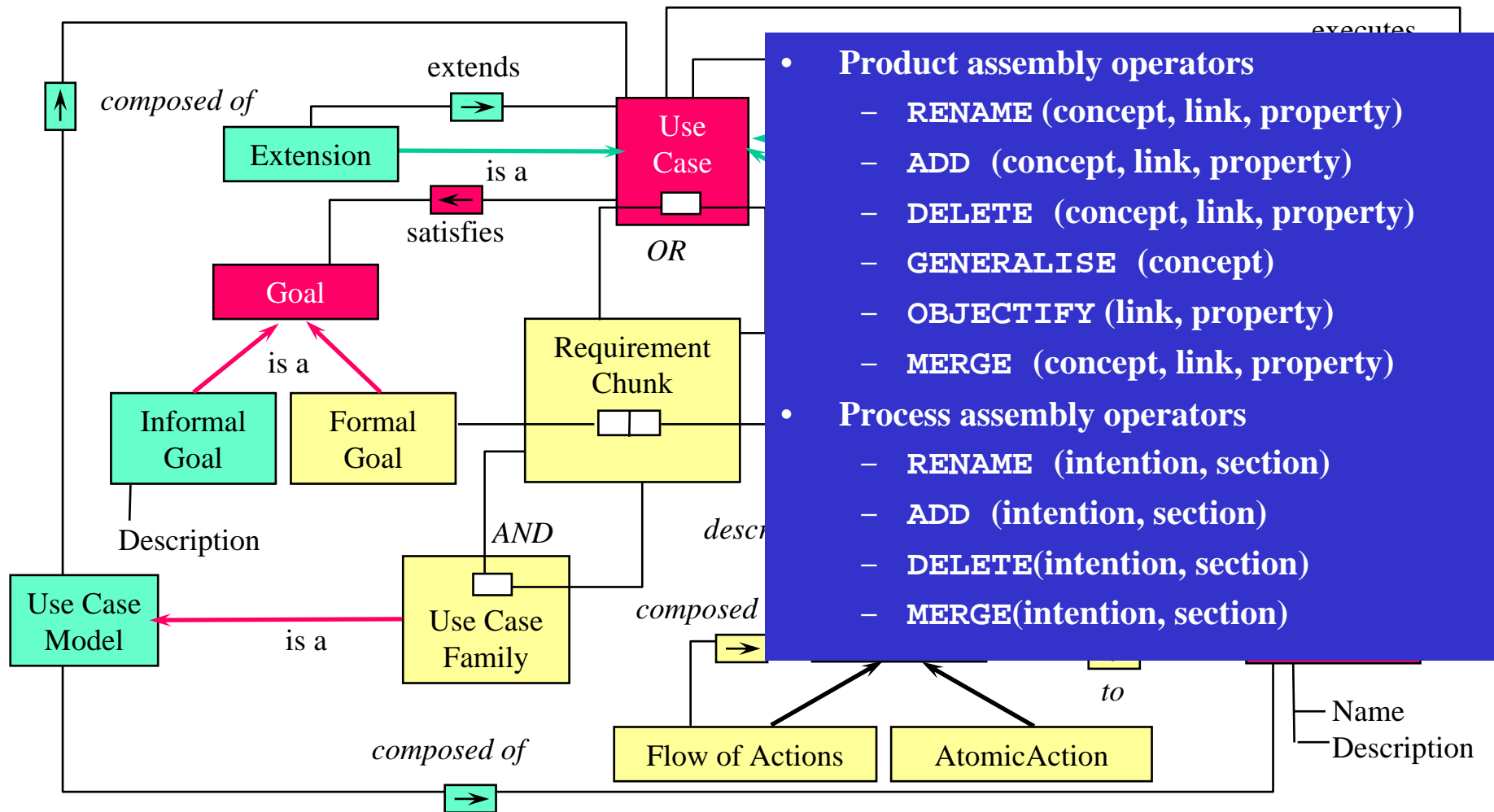
- ♣ Method Assembly by association*: example of product models assembly



ObjectChart, Coleman et al92

Method Engineering Processes

♣ Method Assembly by integration: L'Ecritoire and OOSE



Method Engineering Processes

♣ Less studies on ME processes

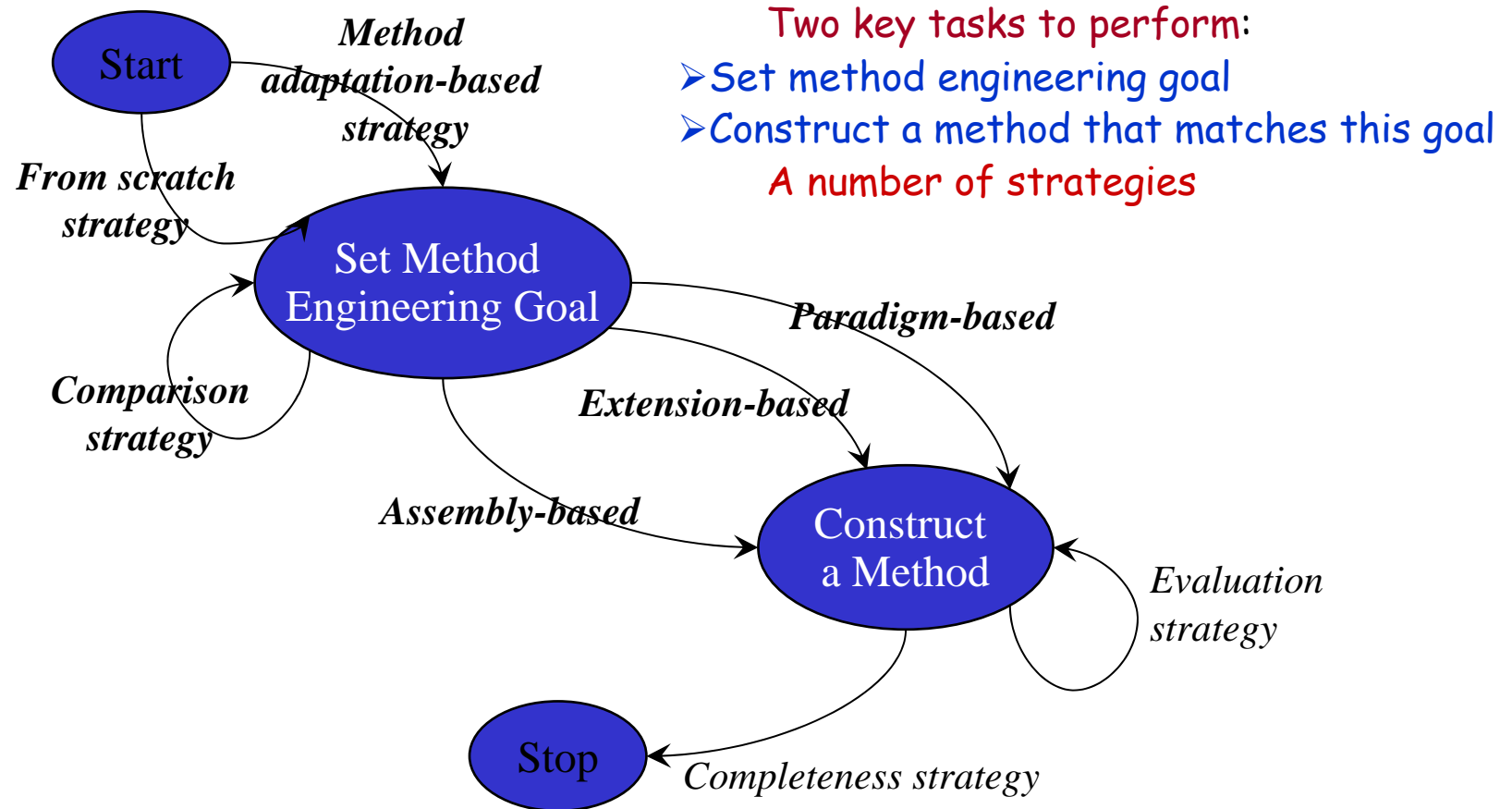
“Towards a life cycle for method engineering “(Gupta, Prakash 2007)

Stage	Process	Input	Output
Requirements engineering	Intention matching	Goal of the method To-Be	Intentionally similar methods
Design engineering	Architecture matching	Architectures of intentionally similar methods	Architecturally similar method
Construction engineering	Organization matching	Workflows of architecturally similar methods	Method To-Be

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

Method Engineering Processes

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



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

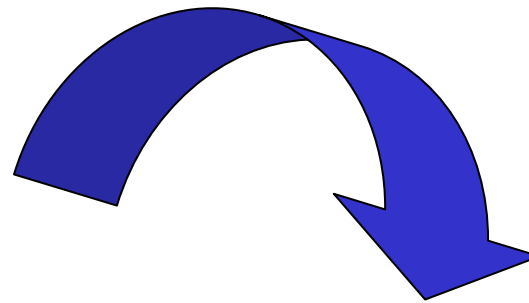
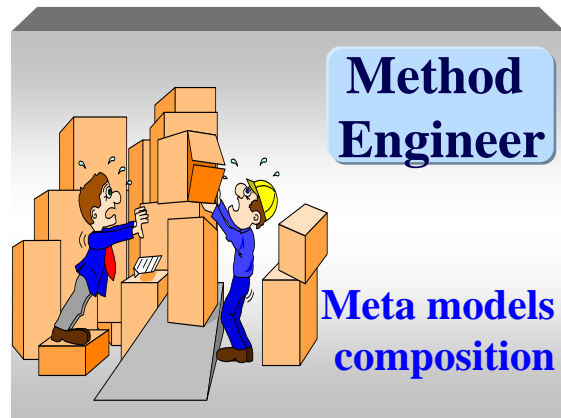
Method Engineering :

♣ 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)

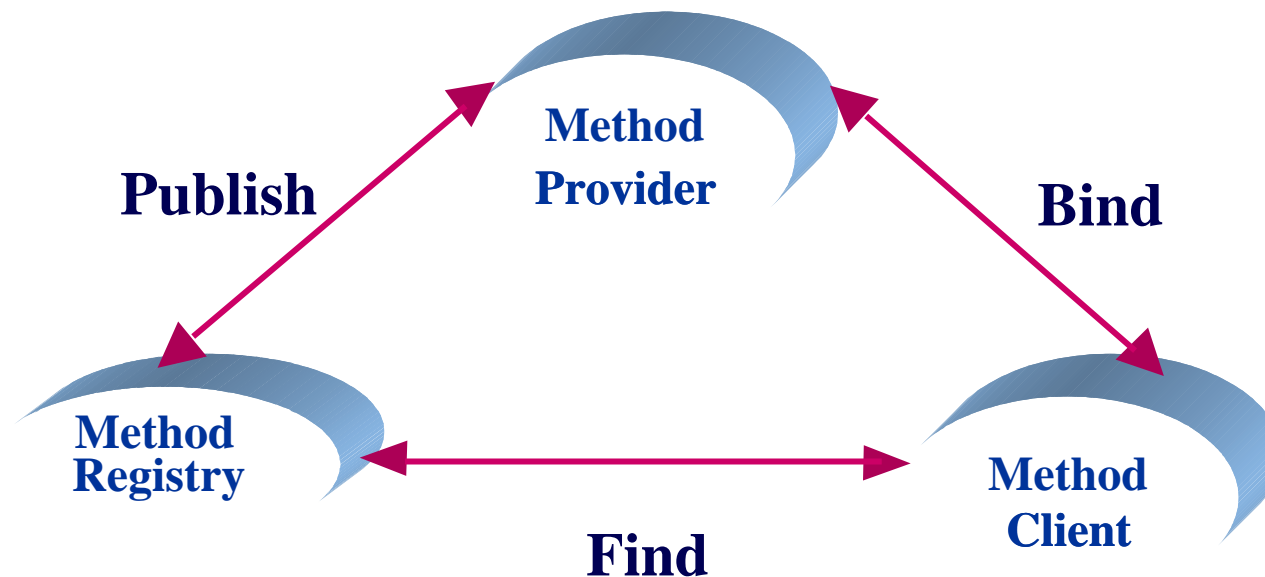
- Emphasize and standardize MP **Interface** descriptions
- **Publish** and make them publicly available
- Provide contextual information to ease MP **finding**
- Facilitate MP **composition**



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



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

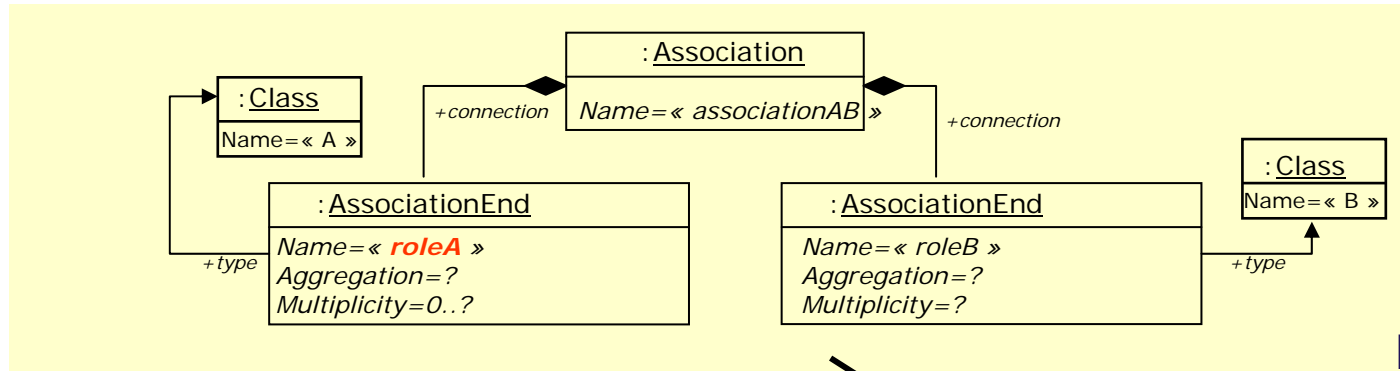
StockQuoteService

GetLastTradePrice
(LastTradePriceRequest)
-> (LastTradePrice)

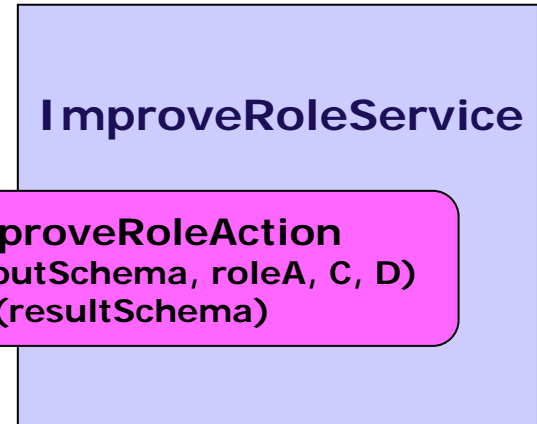
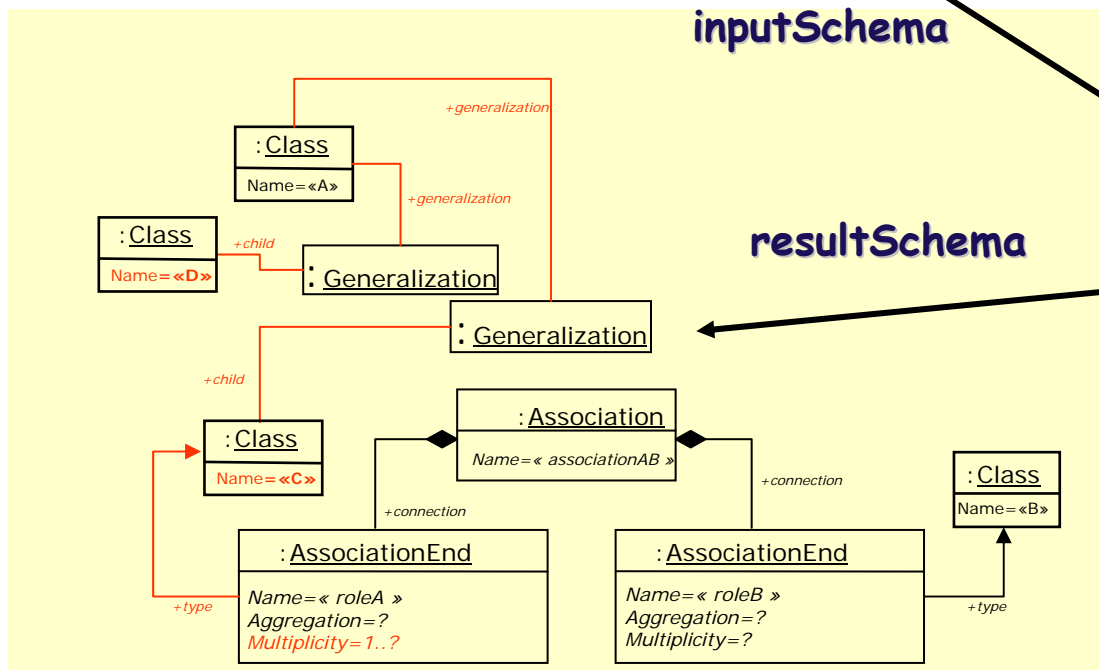
<http://example.com/stockquote>

Method Engineering : MaaS

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



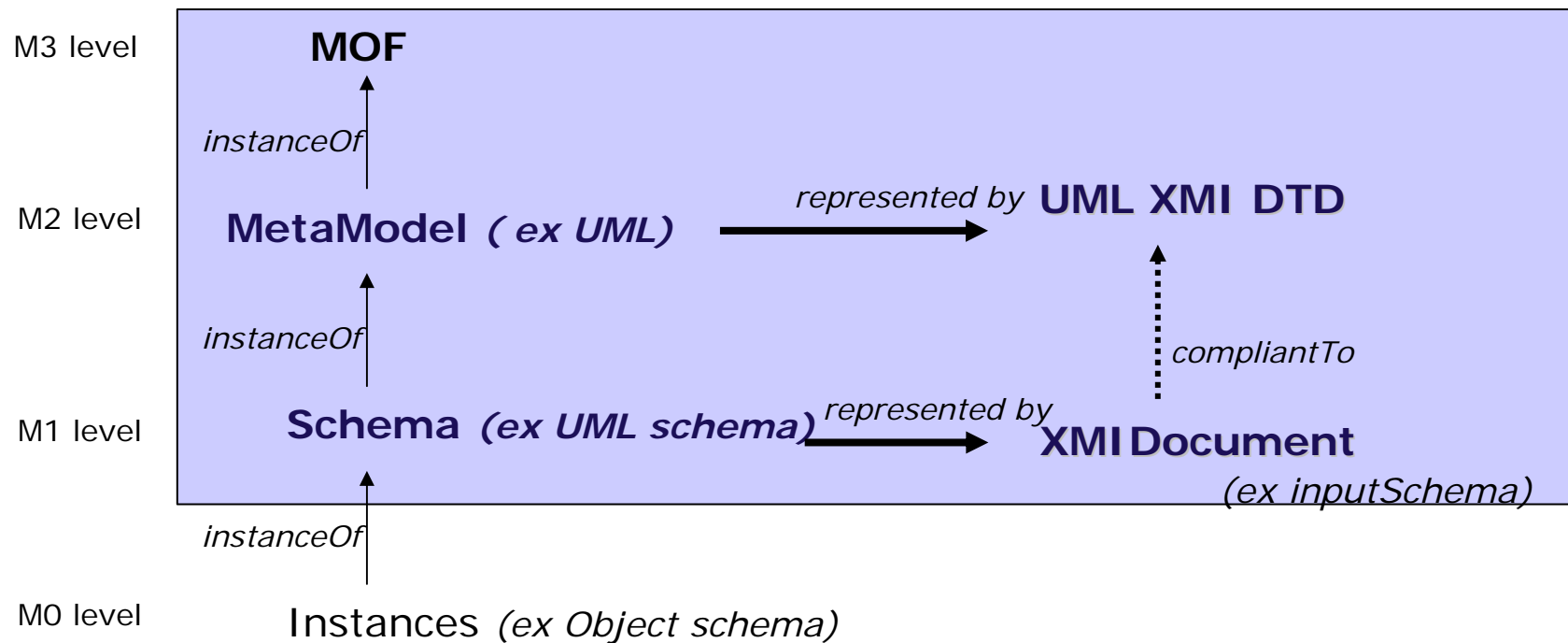
Method Service



Need to understand the meta-model

Method Engineering : MaaS

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



Method Engineering : MaaS

♣ MaaS : An XMI based solution

Types :

ImproveRoleRequest(**inputSchema:XMI Document**,
RoleName : String, ClassWithRole : String, ClassWithoutRole : String)
ImproveRoleResult (**ResultSchema:XMI Document**)

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>

Method service

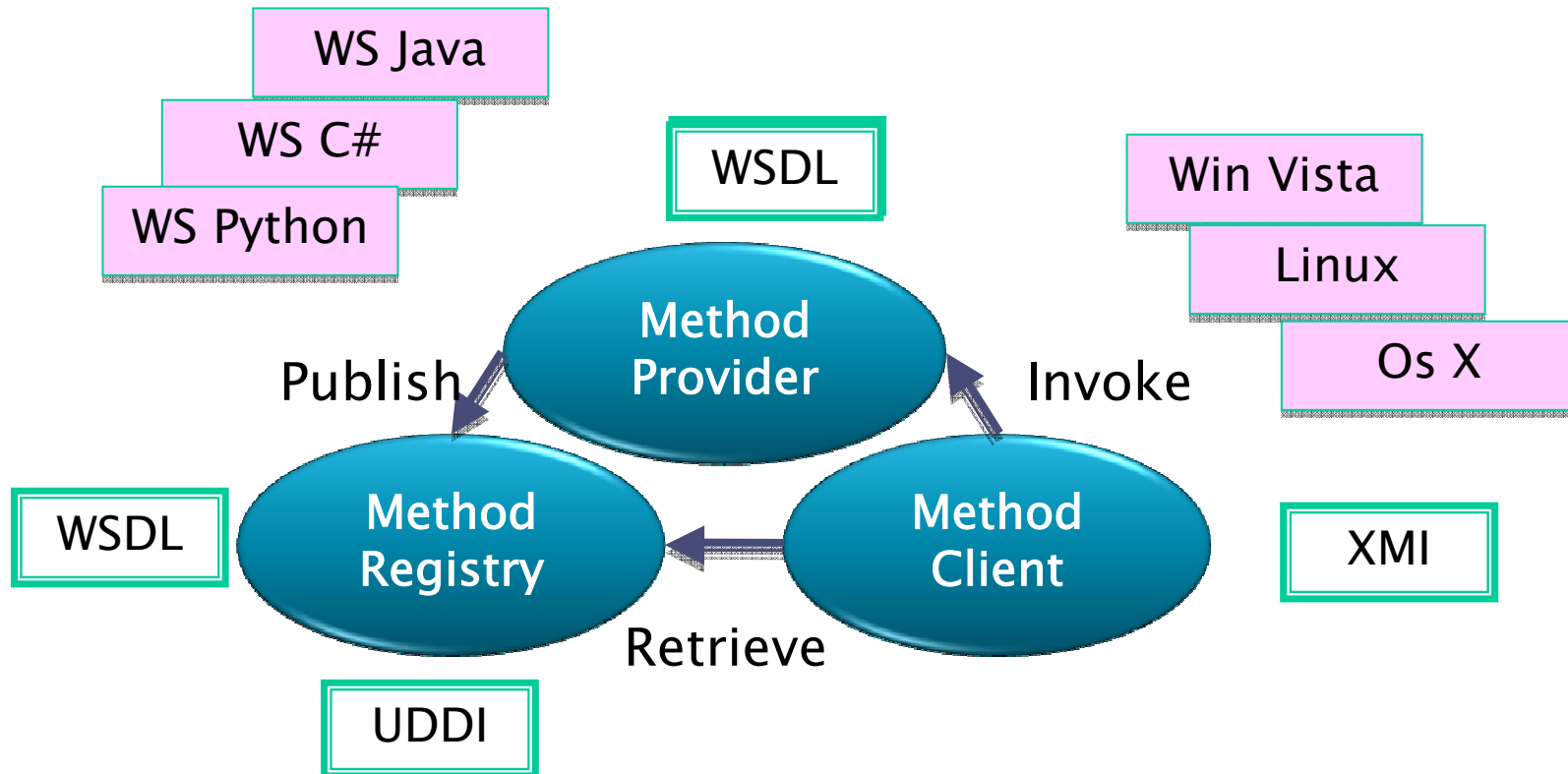
ImproveRoleService

ImproveRoleAction
(**ImproveRoleRequest**)
-> (**ImproveRoleResult**)

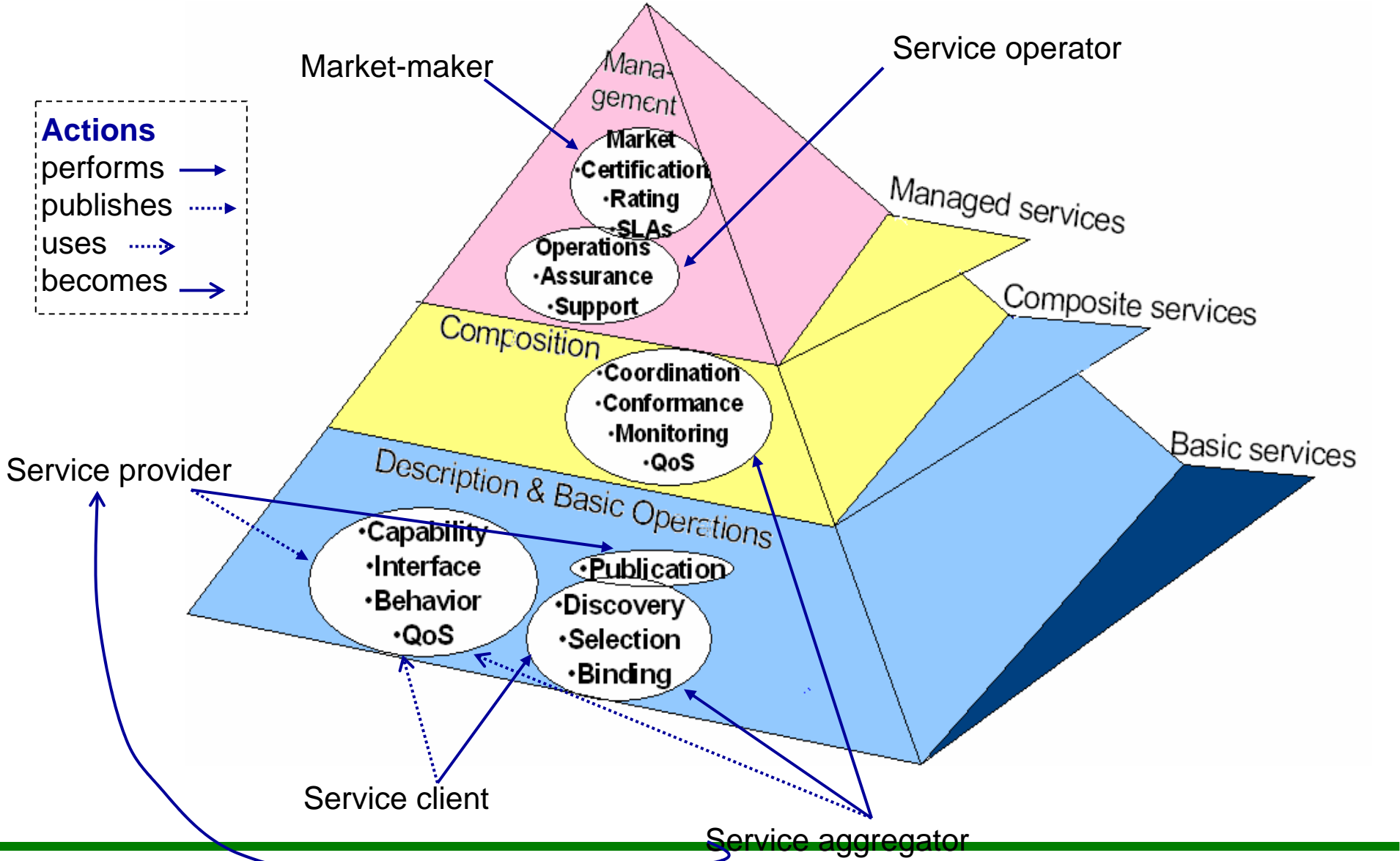
<http://maasexample.crinfo.univ-paris1.fr/improverole>

Method Engineering : MaaS

- ♣ Platform independent & accessible through the Internet



From ME to MaaS Management

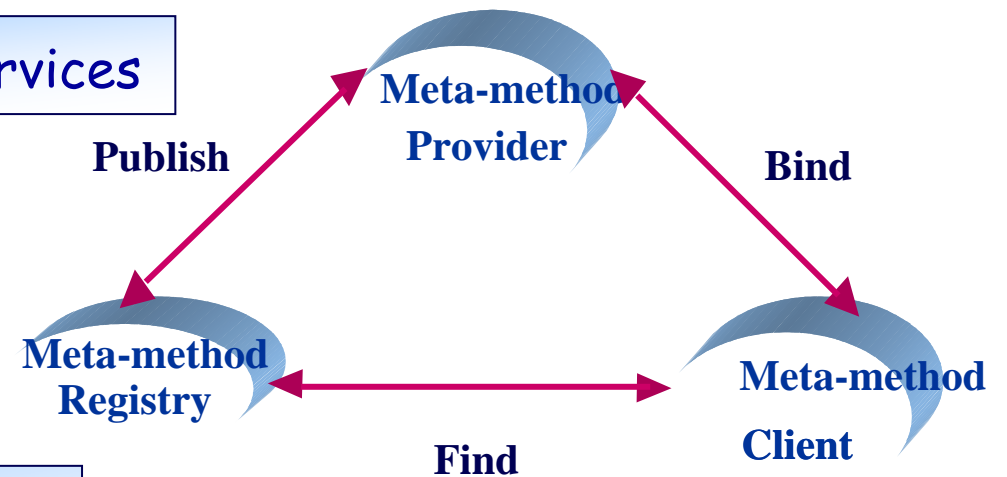


From ME to MaaS Management

♣ Some other issues

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

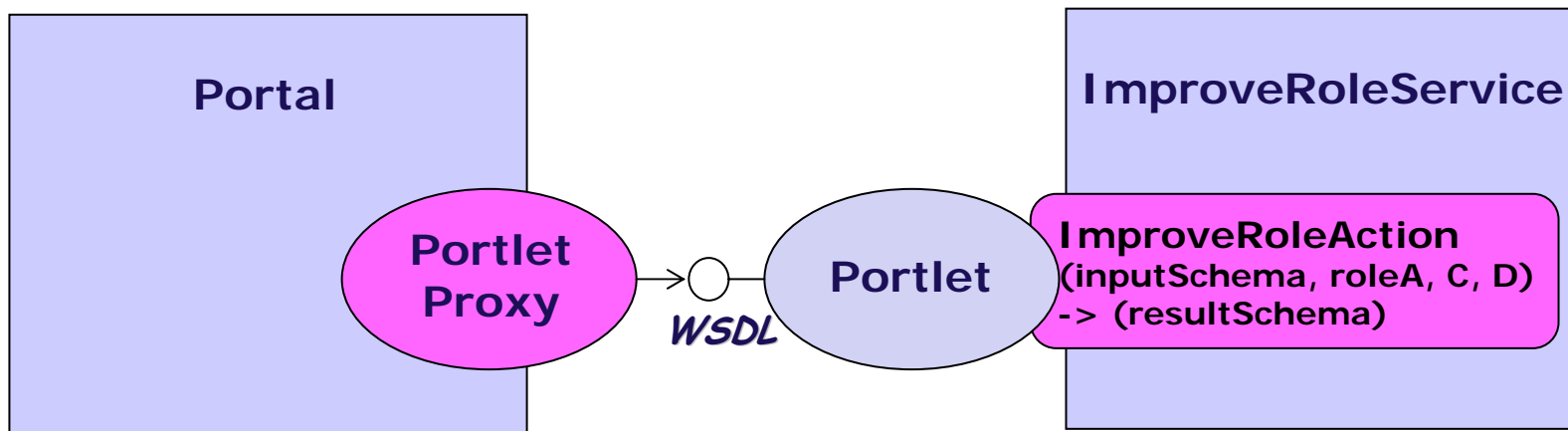
- Providing meta-support as services



- Increasing interaction capabilities in services & meta-services

Method Engineering : MaaS

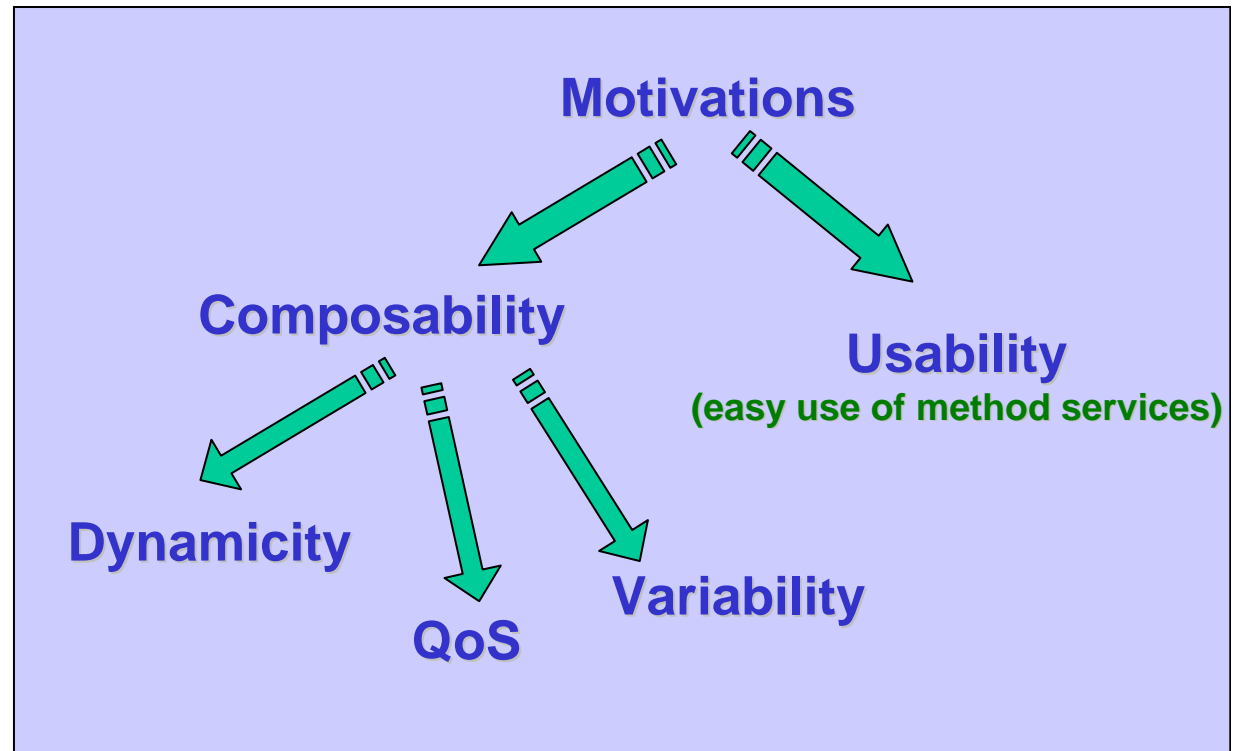
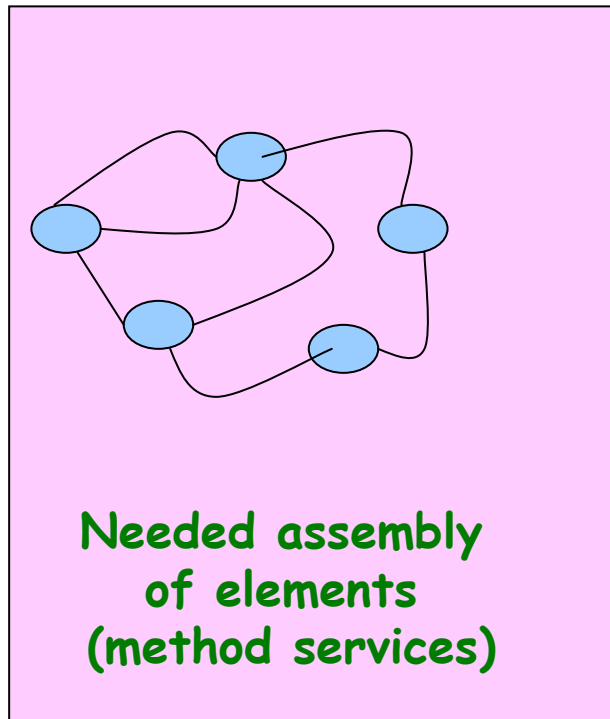
♣ Dealing with interactive advices



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



Thank you for your attention

