

# Cutting The BPM Knot: Why Popular Notations Must Fail

*13.11.2008*

*Andreas Leue, Sphenon GmbH, Hamburg  
www.sphenon.de*



This work is licensed under the Creative Commons Attribution-Share Alike 2.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

# BPM – One Subject, Many Objectives

efficient processes

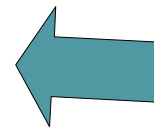


*factor 10*



*factor 1000*

(about 😊)



*factor 10*

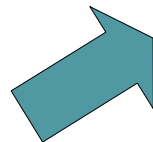


custom-fit IT systems



*factor 10*

optimal task completion



**Competitive advantage by tapping full potential**

# BPM Factors of Success

declared management goal

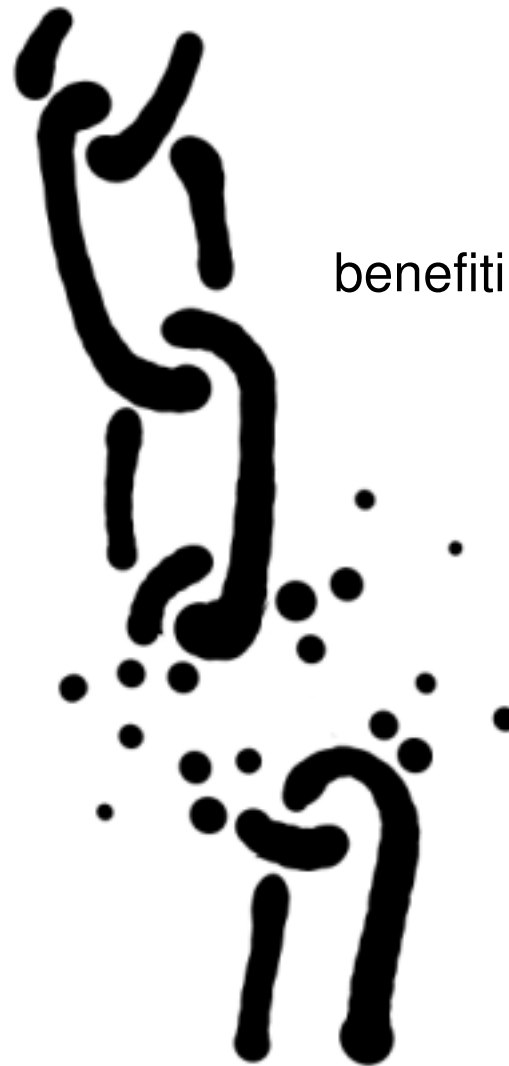
benefiting & participating staff

qualified process architects

effective communications

**suitable modelling language**

economical & working IT



|| → **Starting point: language**

# Model Zoo and Therewith Associated Problems

efficient processes

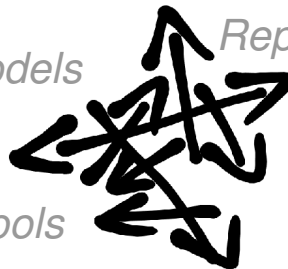


*e.g. Word, Visio*

Management Model

*Metamodels*

*Repositories*



System Model

*e.g. UML, Java*

Process Model

*e.g. BPMN, EPK*



custom-fit IT systems



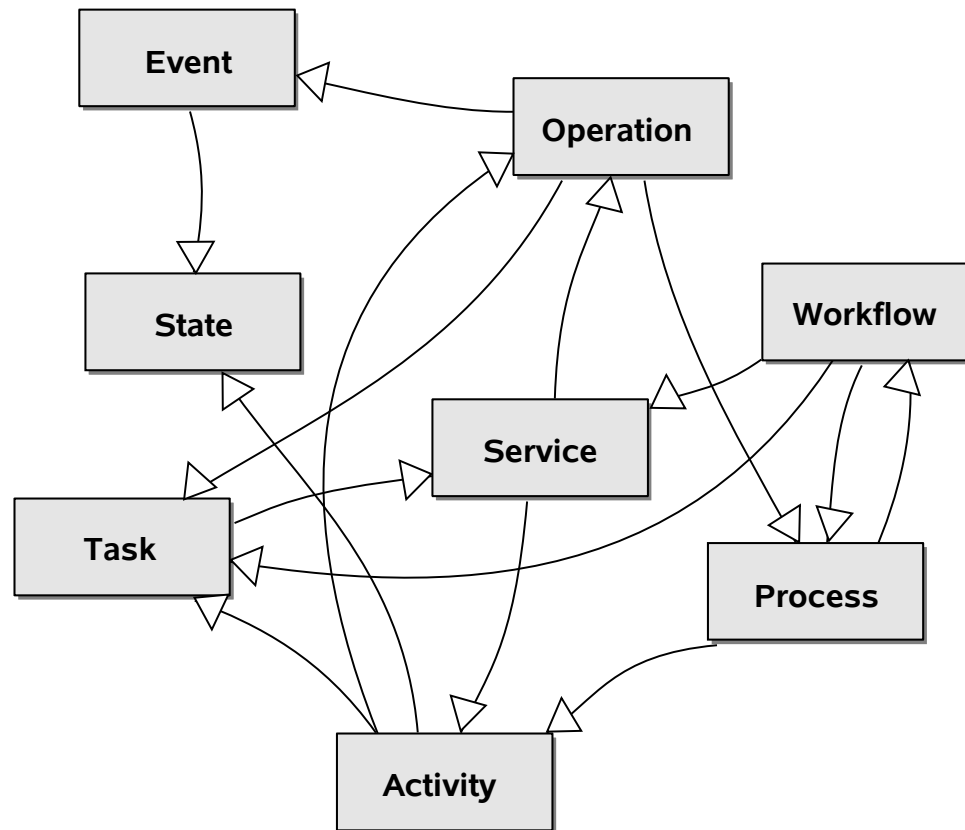
optimal task completion



**Guess: means of expression do not really fit**

# Basic BPM Modelling Terms

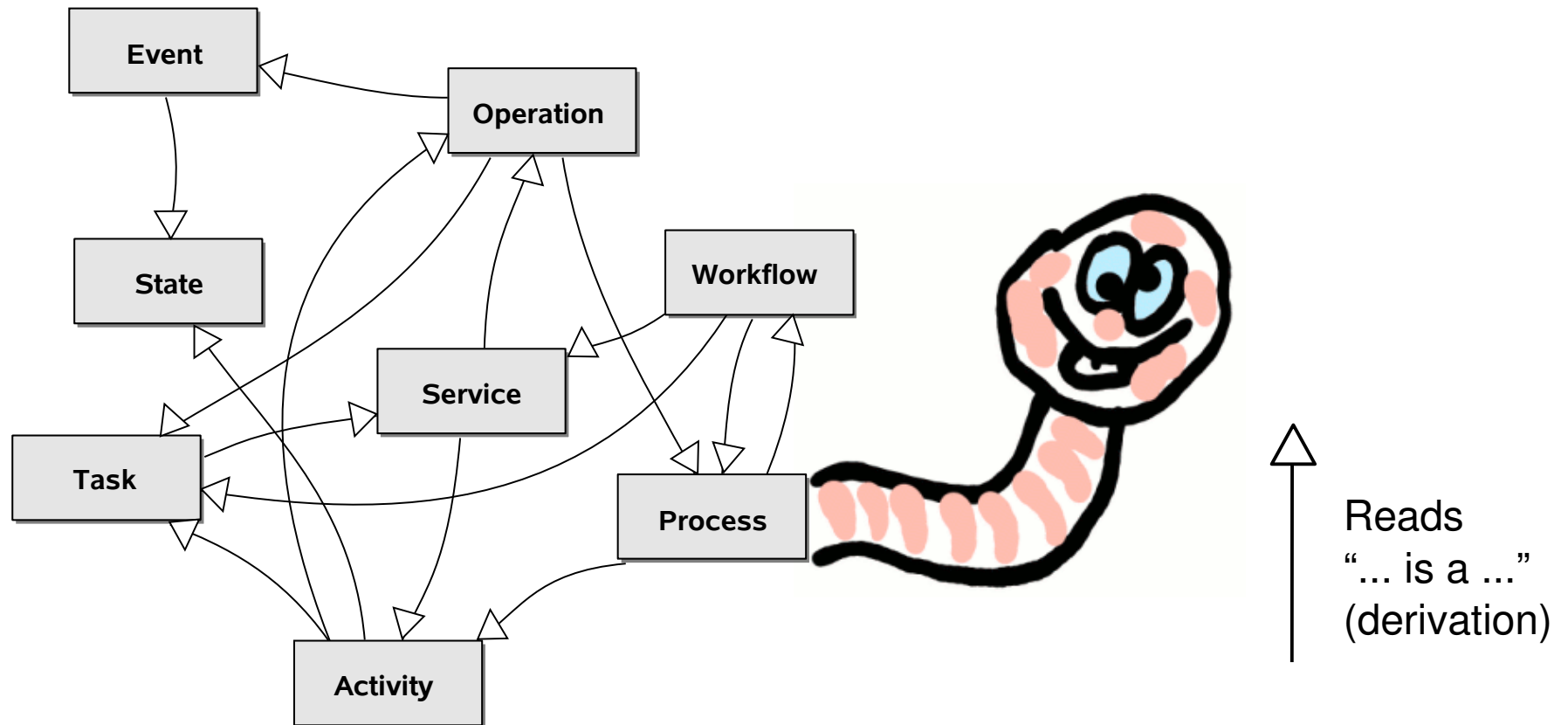
Assembled from several sources (wikipedia.de/.org, specifications and whitepapers about BPMN, BPEL, XPDL, WSDL, UML, EPK, WKD, DFD)



↑  
Reads  
“... is a ...”  
(derivation)

# Basic BPM Modelling Terms

Assembled from several sources (wikipedia.de/.org, specifications and whitepapers about BPMN, BPEL, XPDL, WSDL, UML, EPK, WKD, DFD)



**There's in fact a jinx on it**  
(in german: jinx -> worm)

## **Why Sphenon Is Engaged in This**

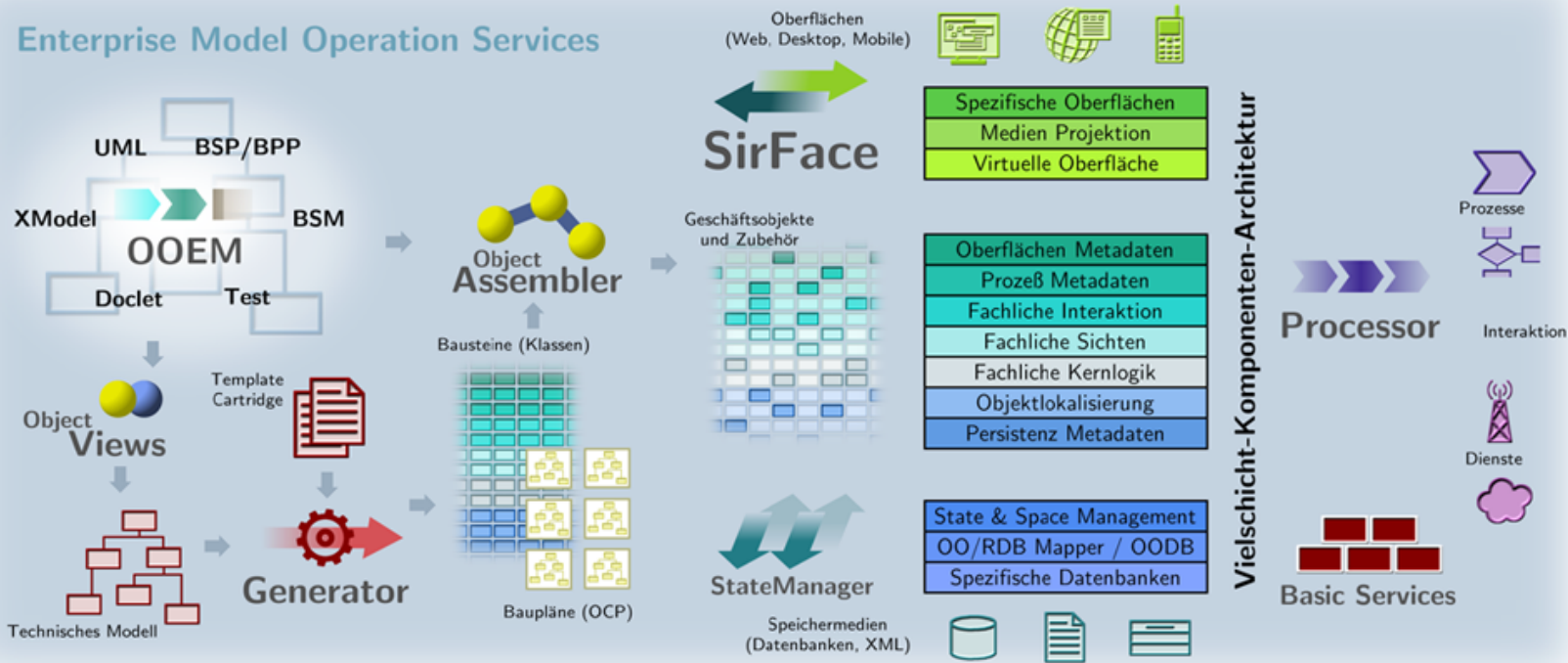
# Enterprise Model Operation Services (EMOS)

since 1990 development of a system for automatic provision of applications

model in



## Enterprise Model Operation Services



BPM integration since ca. 2000

application out

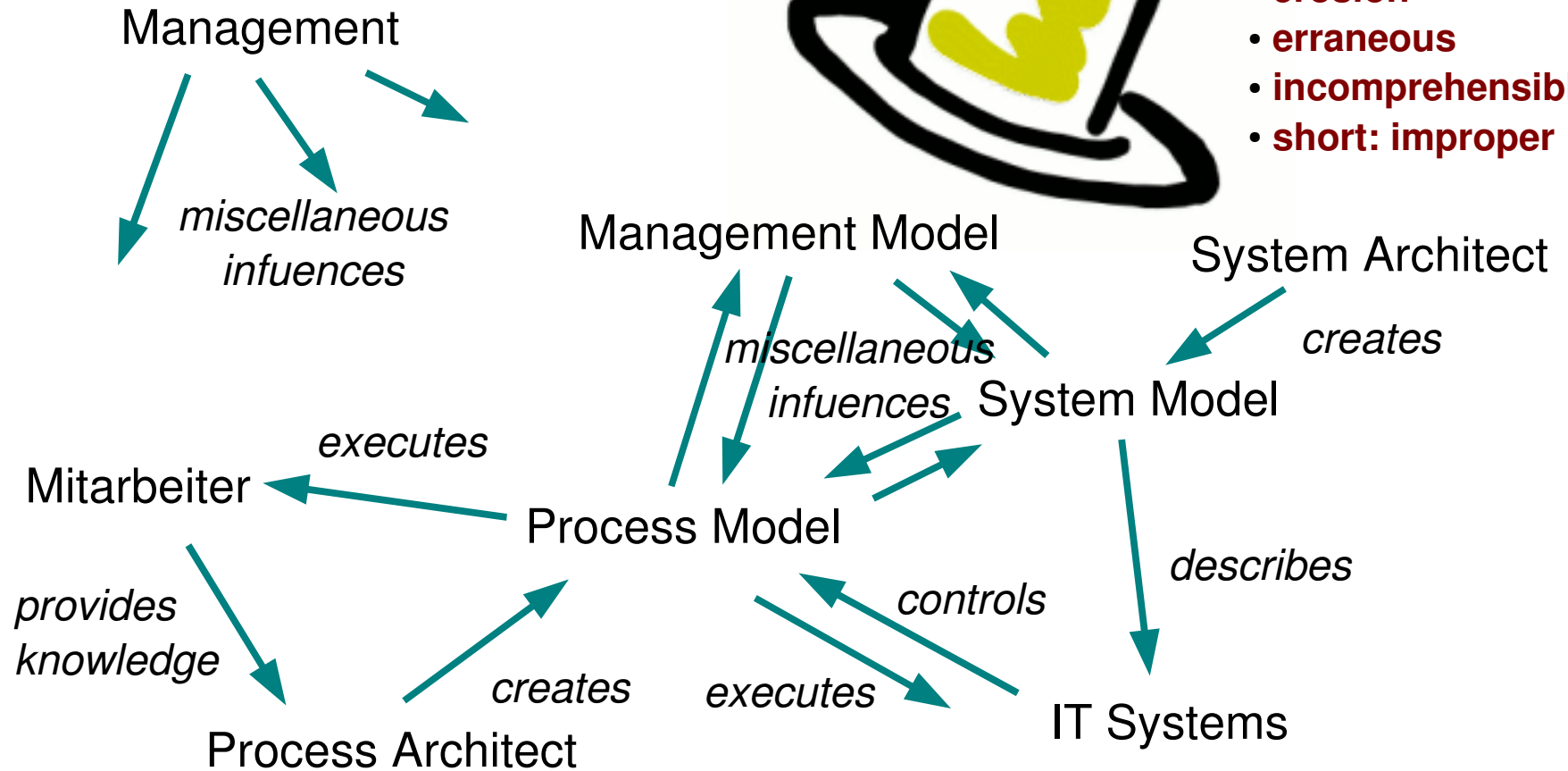


# Dead And Living Models

# Model Diversity

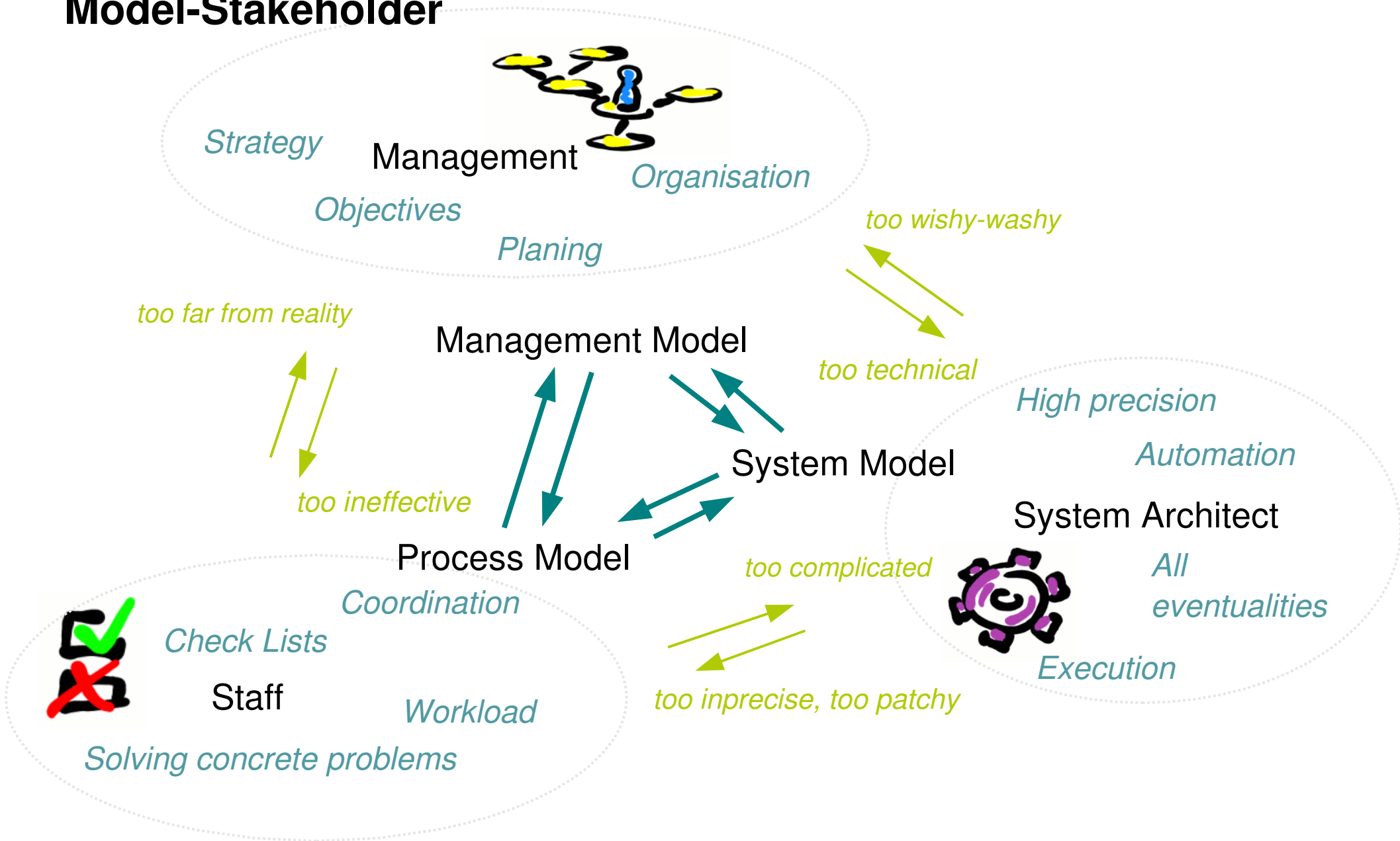


- **redundant**
- **inconsistent**
- **erosion**
- **erraneous**
- **incomprehensible**
- **short: improper**



⇒ **Does not work in the long run**

# Model-Stakeholder



➡ **At least 3 clearly different views**

# Conditions for A Living Model

Living = always up to date and maintained

a) rigid discipline -> well, possible...

b) Integration by usage and necessity:

- **tight coupling to enterprise planning:**

- ➔ Comprehensibility
- ➔ Effectivity
- ➔ Abstracting description

- **tight coupling to every-day tasks:**

- ➔ Usage by staff
- ➔ Feedback from staff

- **tight coupling to IT:**

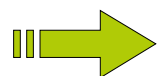
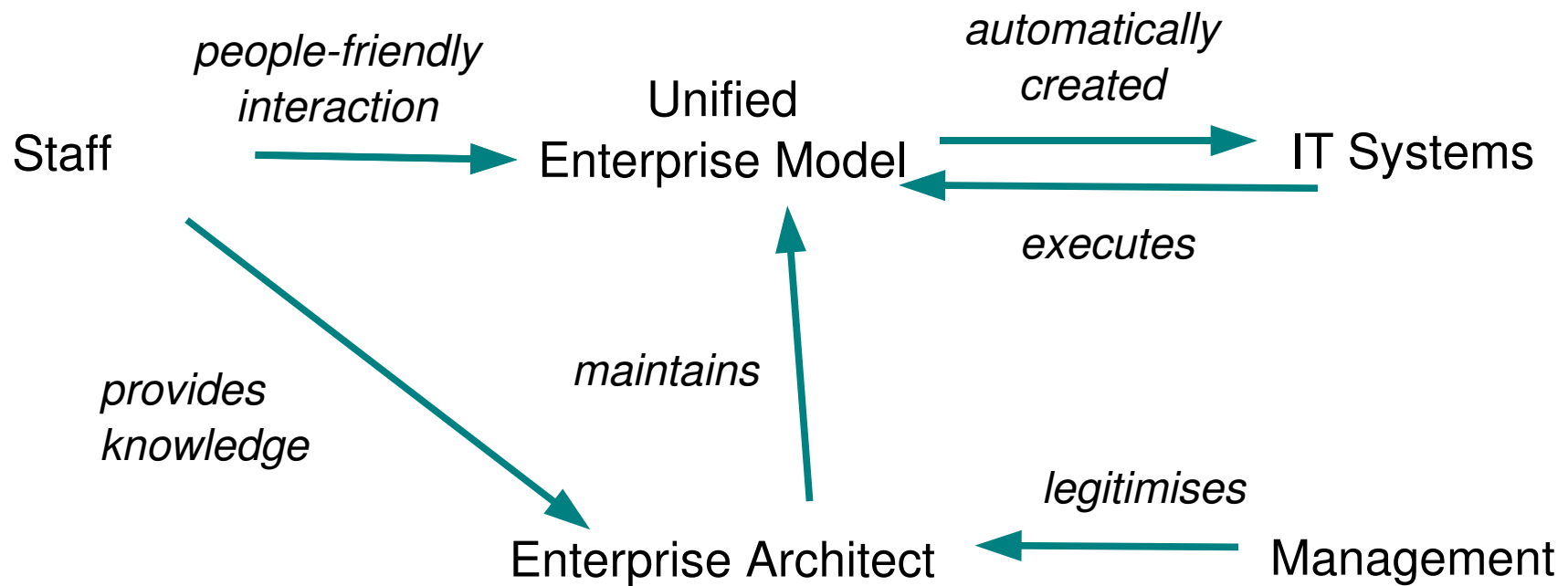
- ➔ Completeness and precision
- ➔ automatic creation of software
- ➔ Managed and controlled by software

⇒ **Precisely 1 used model with abstracting hierarchy**

# Unified Enterprise Model

UML 3.0 ?

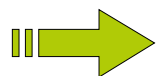
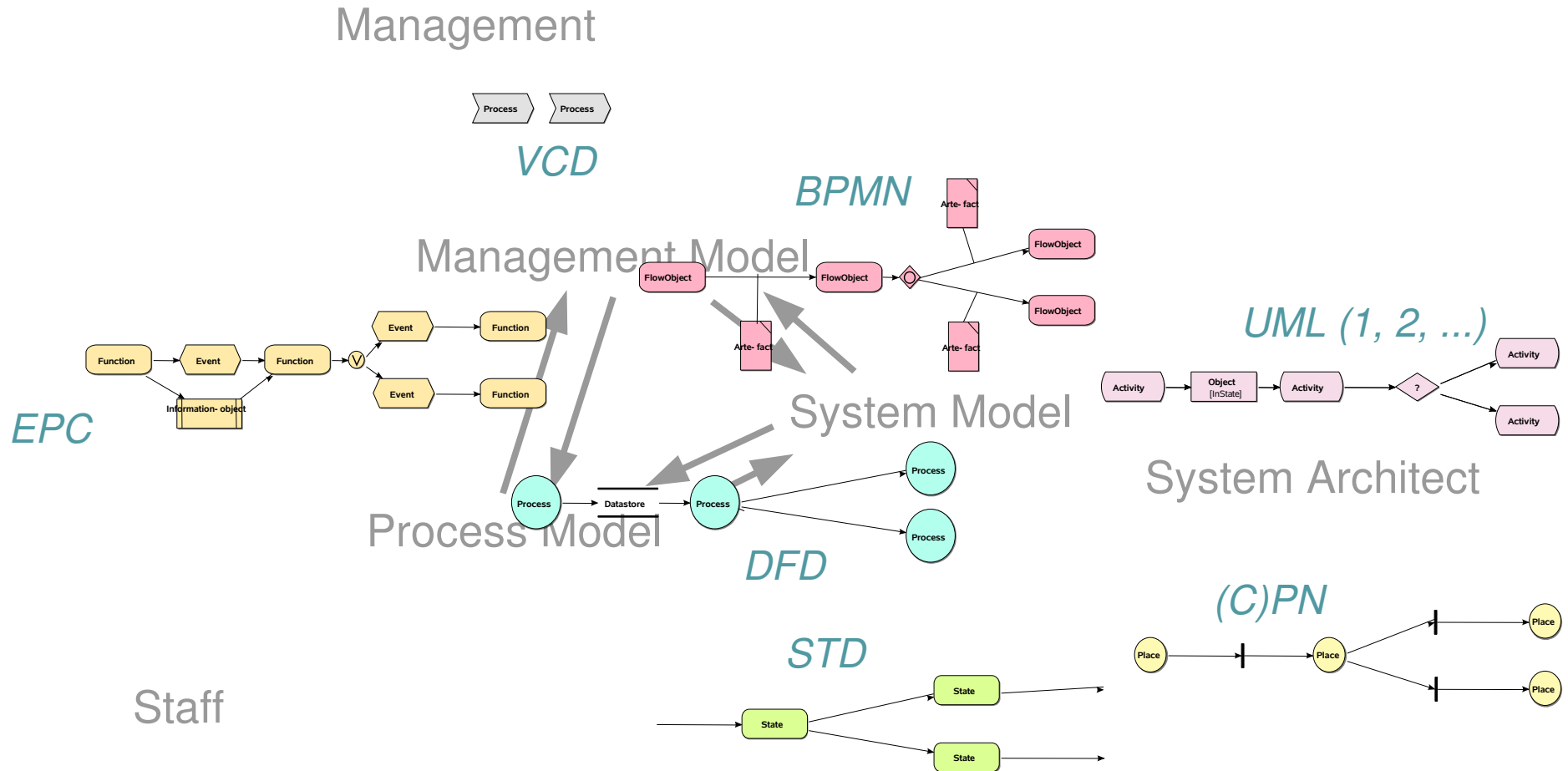
OOEM ?



**Goal – overcome babylonic circumstances**

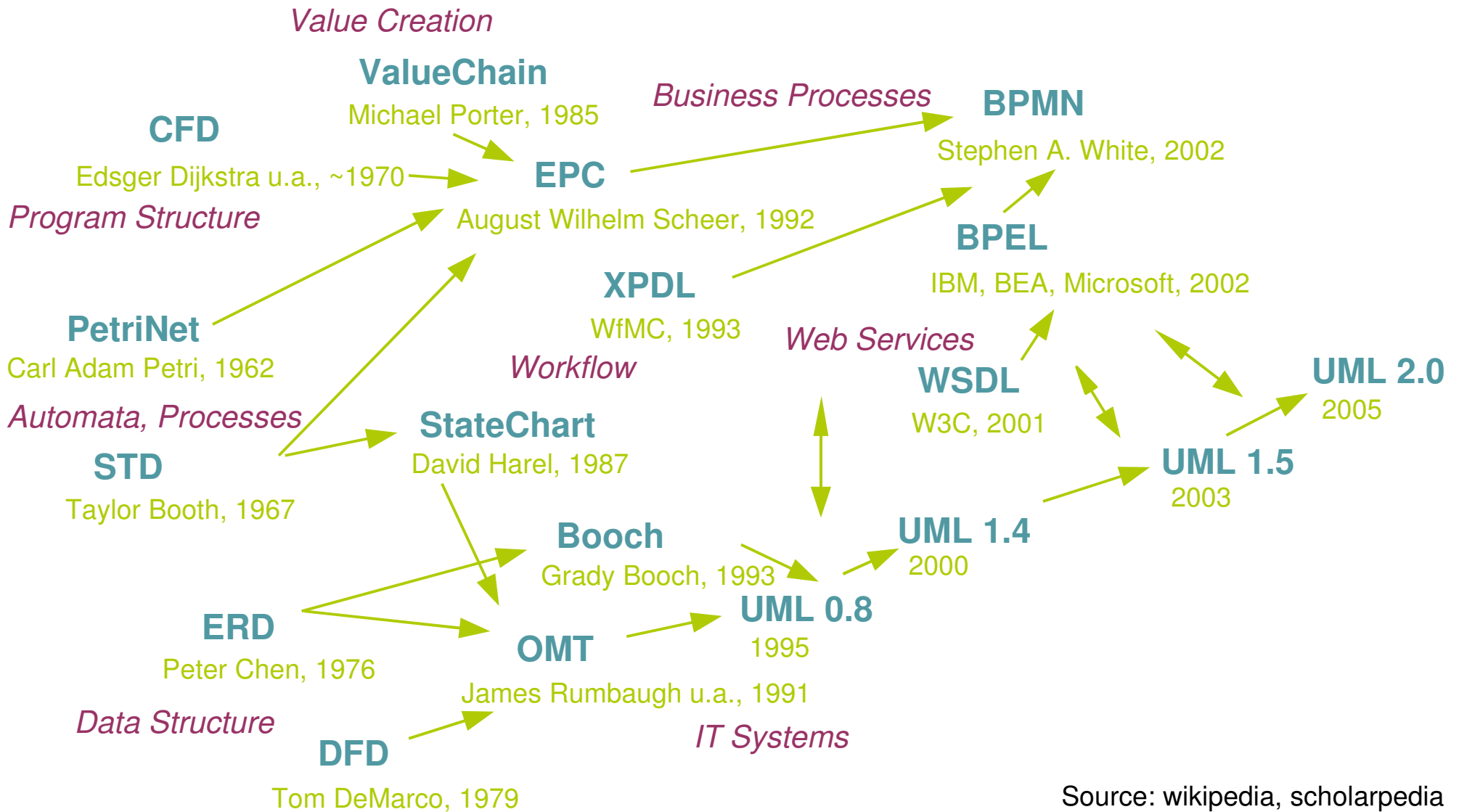
# Modelling Languages

# Different Affinity to Modelling Languages



Lukewarm connected siloed solutions

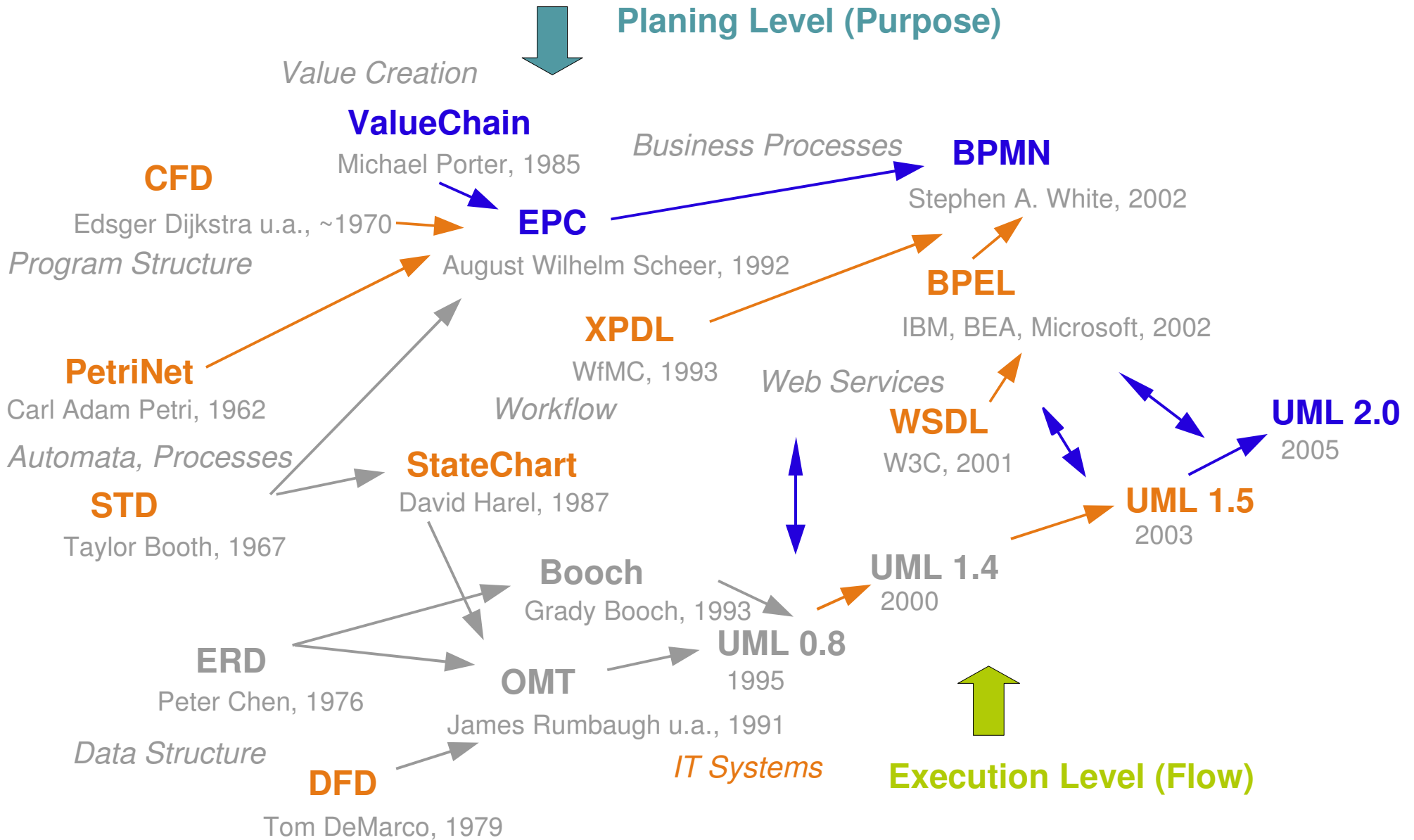
# Evolution of BPM And IT Modelling Languages



Source: wikipedia, scholarpedia



# Evolution of BPM And IT Modelling Languages

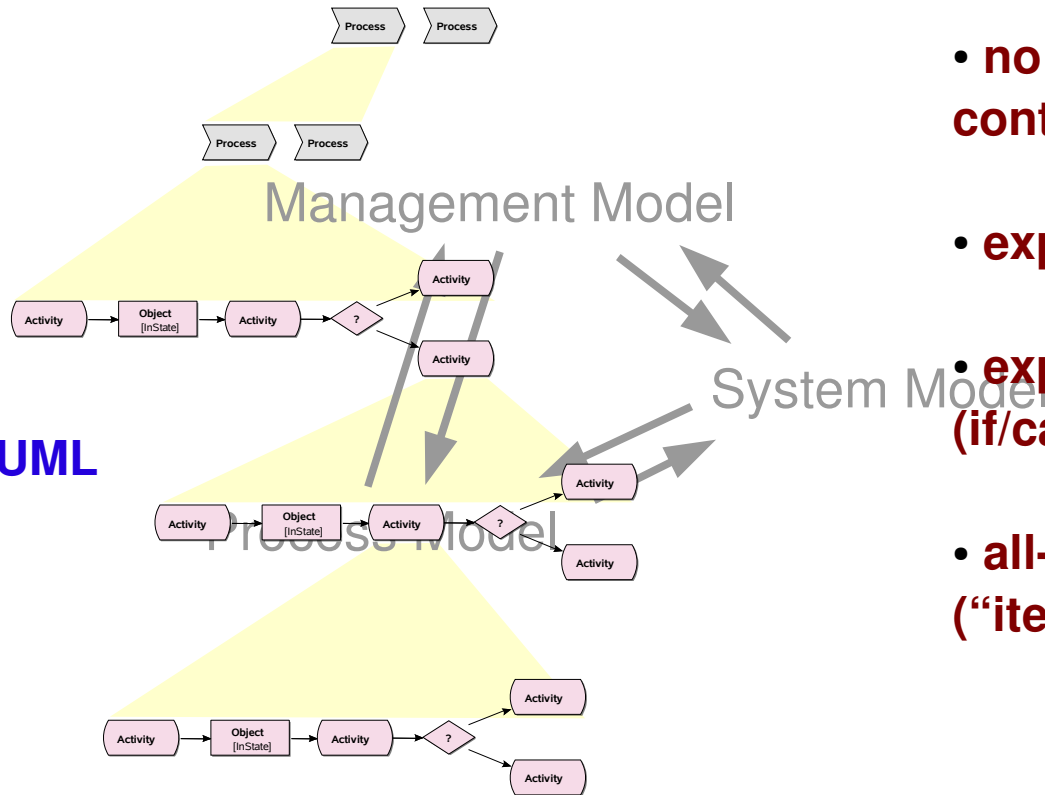


➡ **Concrete application requires floworiented notations**

# Flow Orientation (Control Flow, Token Semantics)

Planing Level (Purpose)

VCD



BPMN, EPC, UML

Execution Level (Flow)

- primarily procedural decomposition
- no abstraction regarding control flow
- explicit exception handling
- explicit case differentiation (if/case, “Rules”)
- all-or-nothing-semantics (“iterative models”)



Flow als primary structure... déjà-vu: 1990, OO...

**OO + BPM ...**

# Objects in BPM



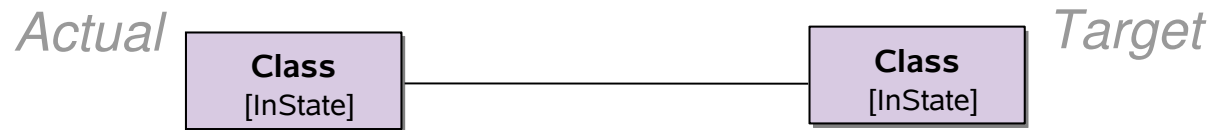
1. Material one operates on

2. Activities as methods

⇒ Not yet OO modelling of “BPM” domain

# Objects in BPM

## Planing Level (Purpose)



## Execution Level (Flow)

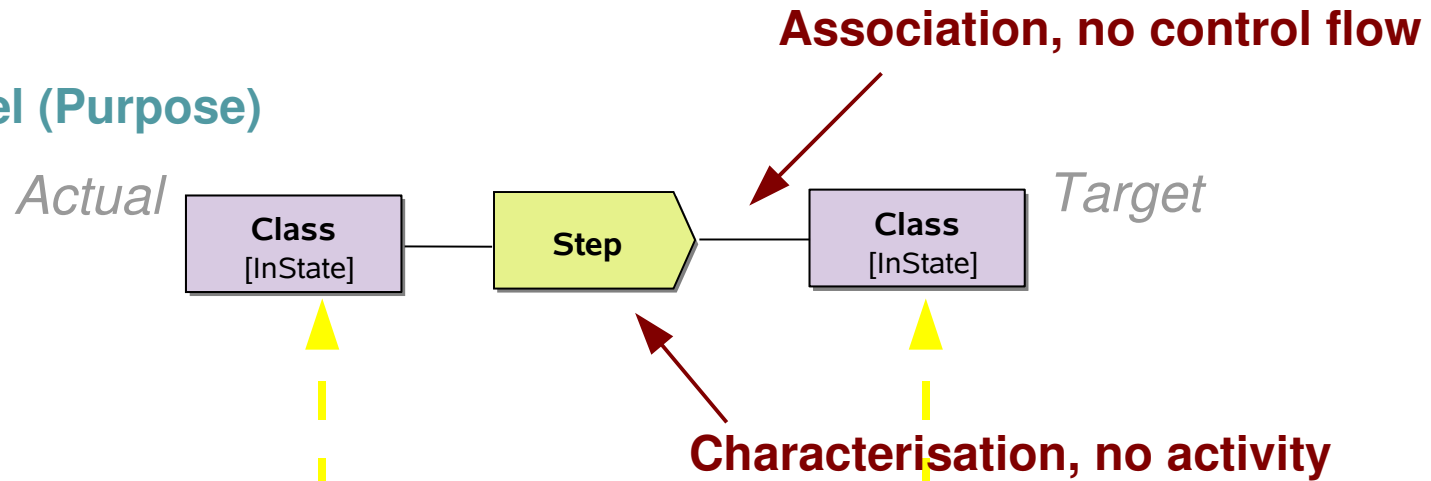


1. Material one operates on

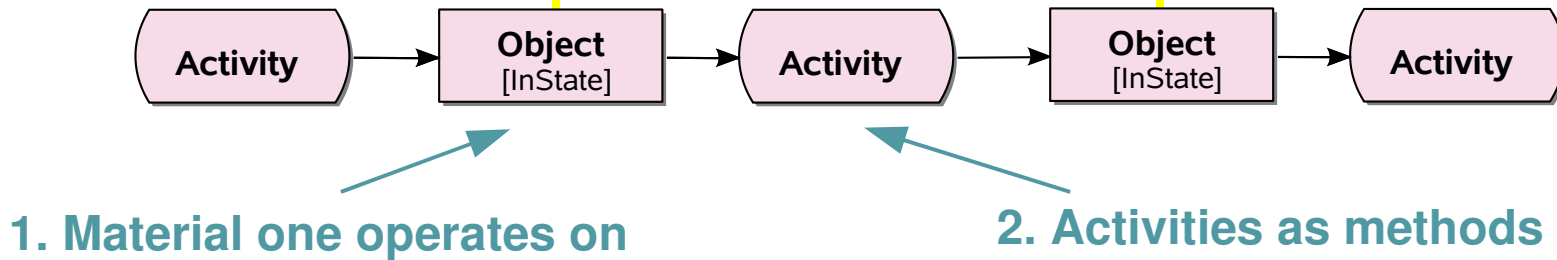
2. Activities as methods

# Objects in BPM

## Planing Level (Purpose)

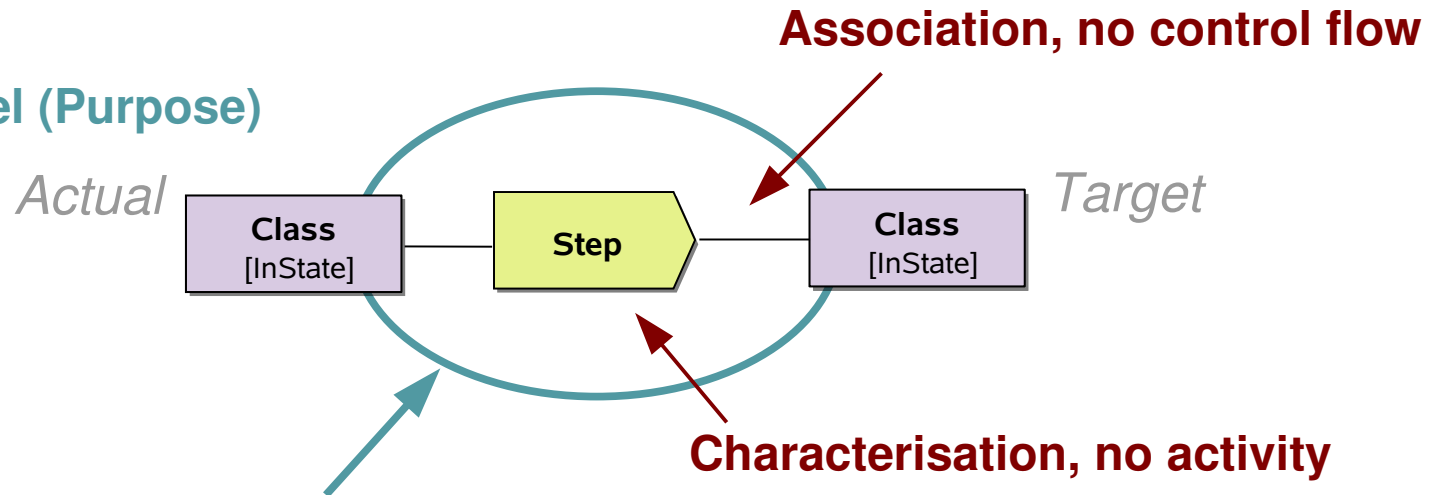


## Execution Level (Flow)



# Objects in BPM

## Planing Level (Purpose)



## Execution Level (Flow)



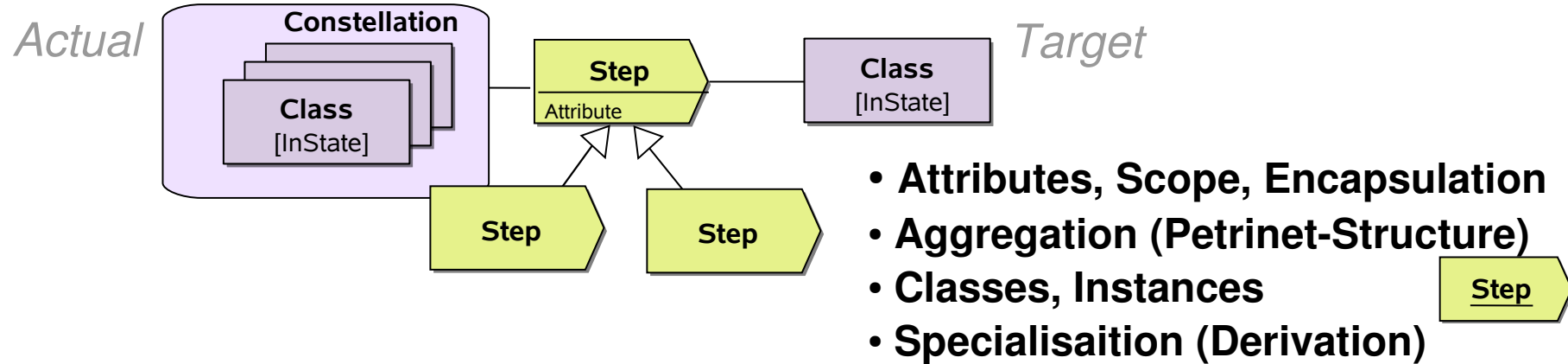
**1. Material one operates on**

**2. Activities as methods**

➡ **Approach “Steps = Objects” - is it of use?**

# Objects in BPM

## Planing Level (Purpose)



## Execution Level (Flow)

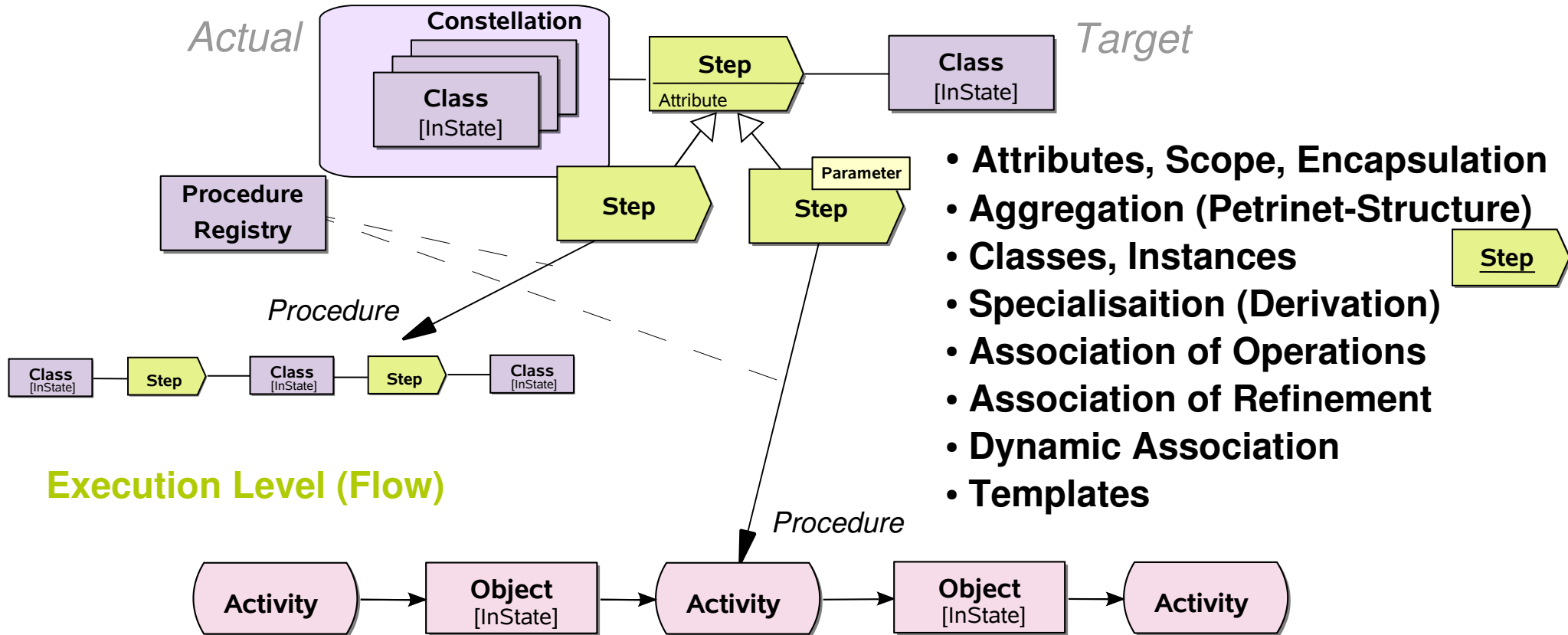




# Objects in BPM

## Planing Level (Purpose)

*Actual* Constellation *Target*



- Attributes, Scope, Encapsulation
- Aggregation (Petri-net-Structure)
- Classes, Instances
- Specialisation (Derivation)
- Association of Operations
- Association of Refinement
- Dynamic Association
- Templates

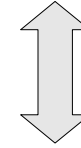
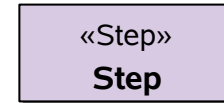
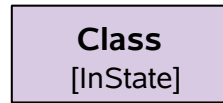
➡ The whole OO toolkit is available

# Example

# Notation OEM

Object Oriented Enterprise Modelling - [www.oem.org](http://www.oem.org)  
(OEM = UML + XModel + OOBPM + Doclet + Test + Spaces)

As UML Profile:



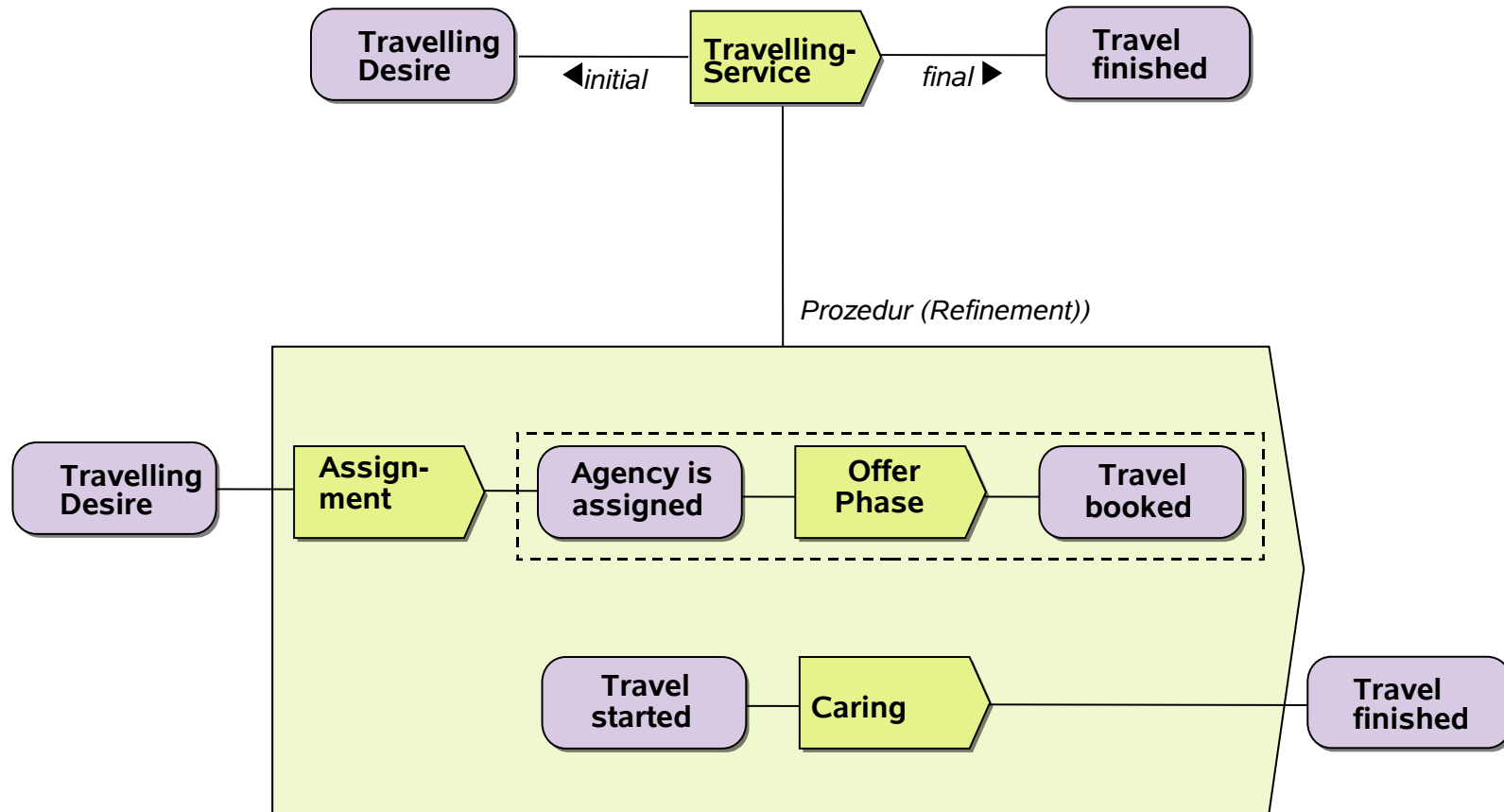
With Stereotype-Symbols:



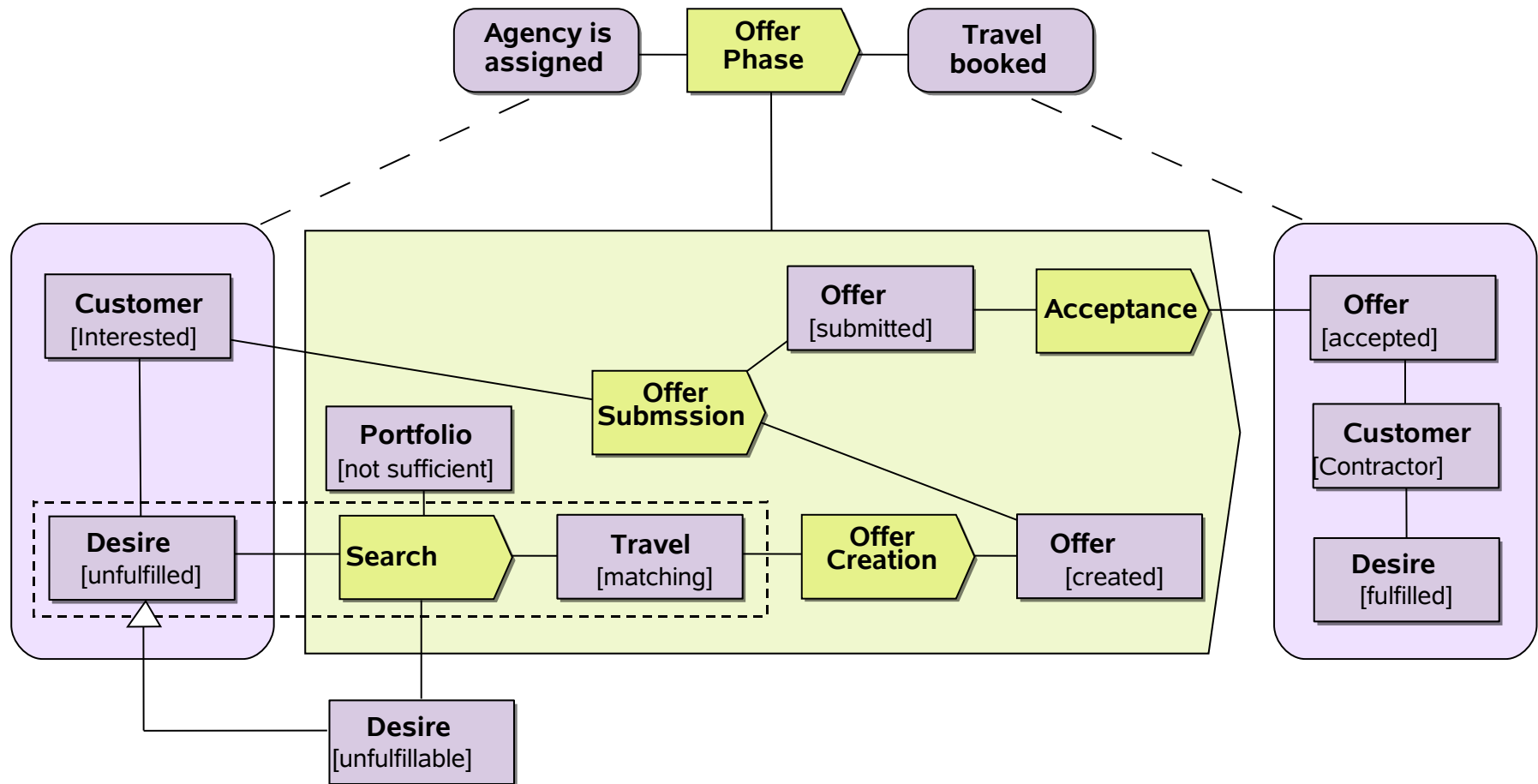
# Example: Travellingservice



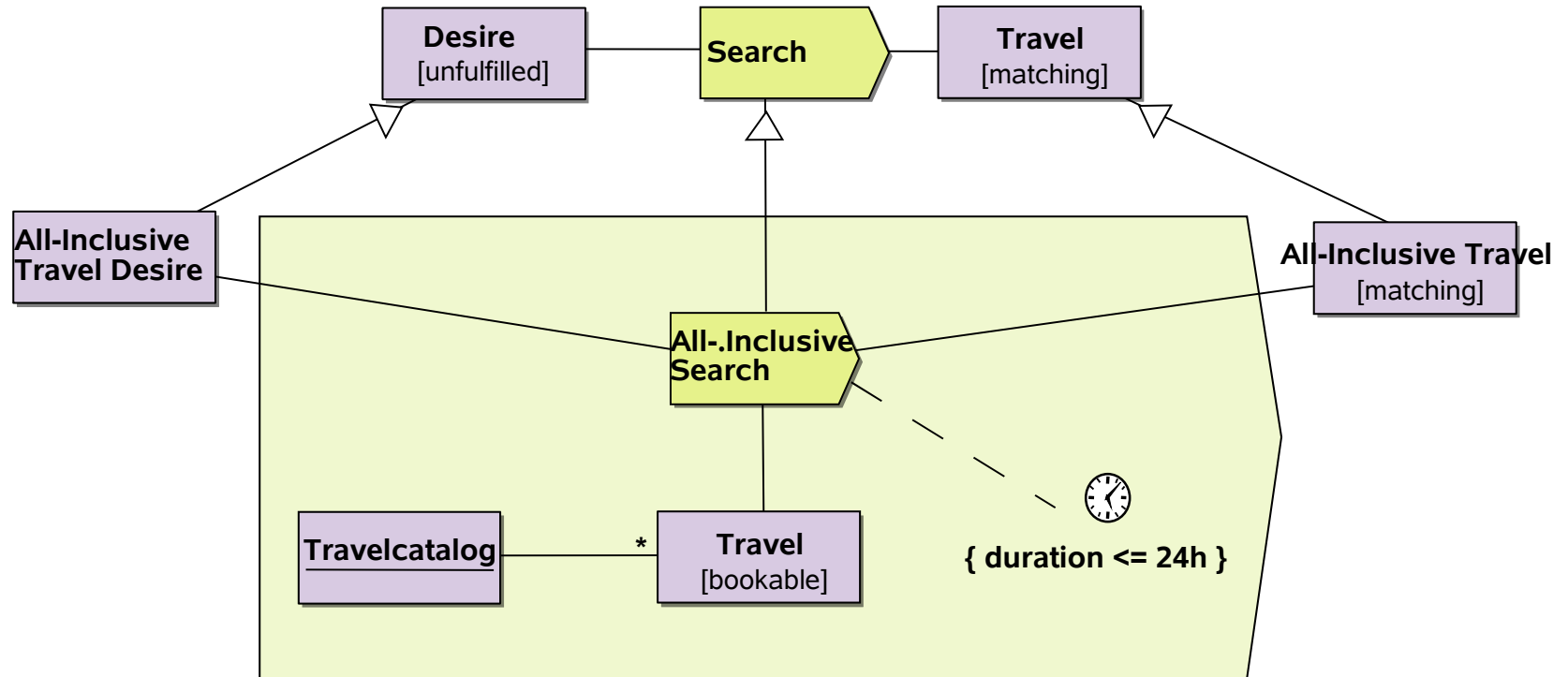
# Example: Travellingservice



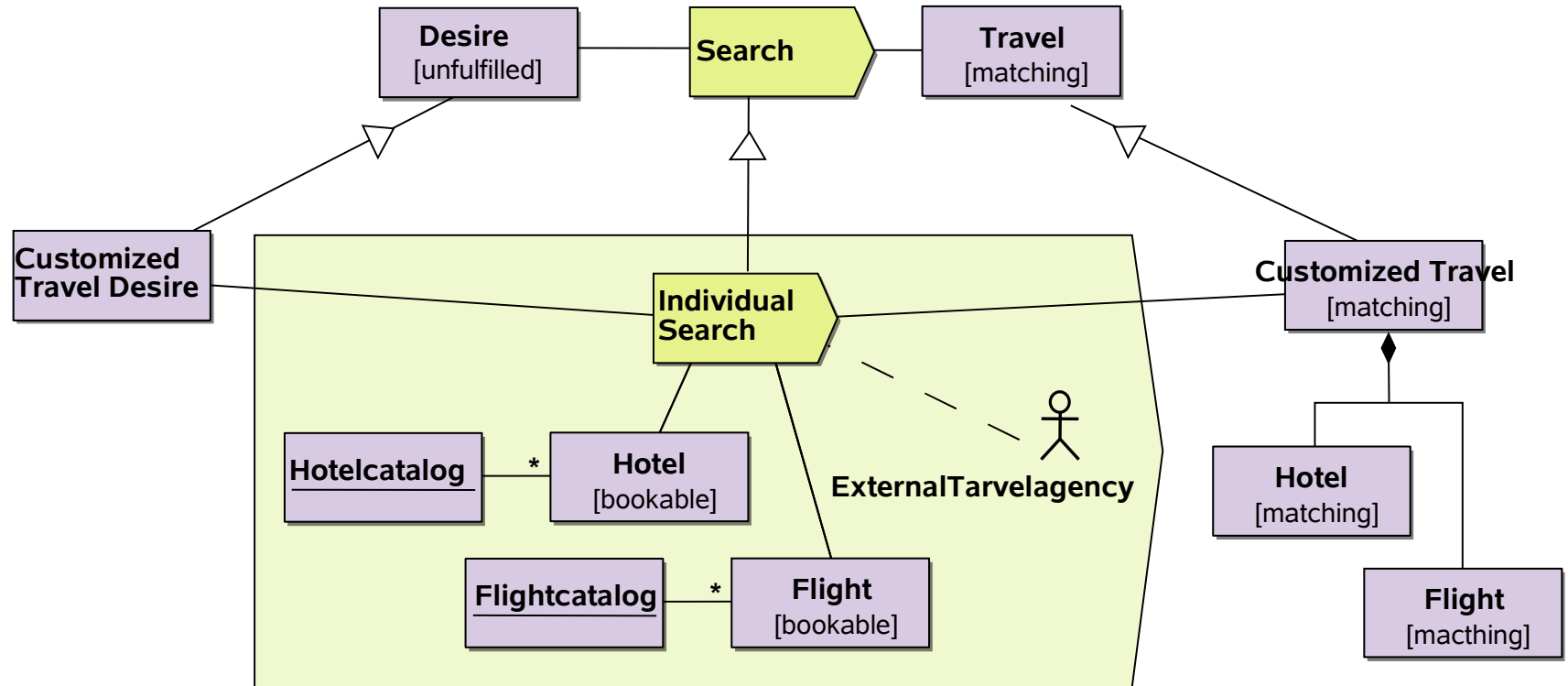
# Example: Travellingservice



# Example: Travellingservice

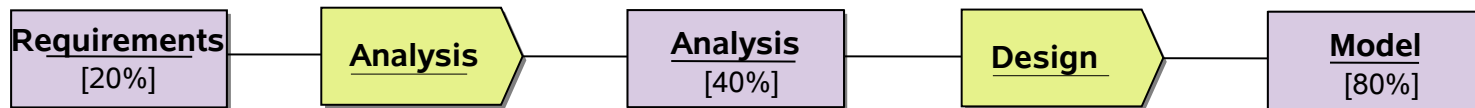


# Example: Travellingservice





# Example: Development Process



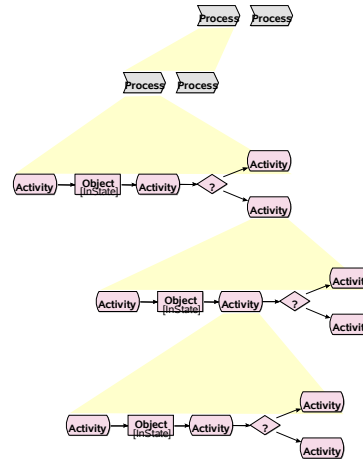
# Summary



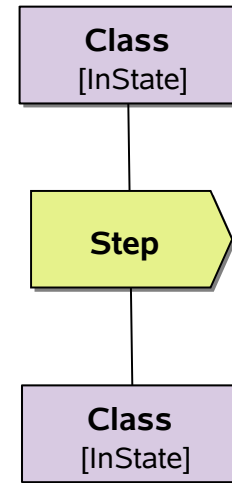
BPM Factors of Success



Model Diversity

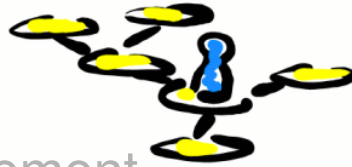


Flow Orientation



Object Orientation

# OO-BPM Models



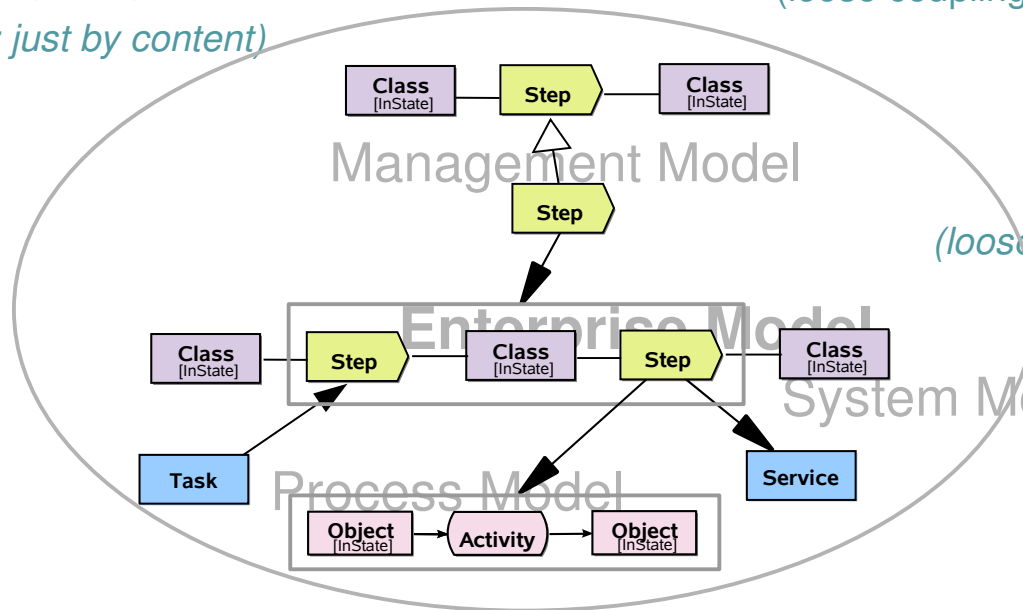
Management

## Inheritance

(loose coupling by abstraction)

## No Control Flow

(loose coupling: just by content)



## Refinement

(loose coupling by dynamic Association)

System Architect



## IT Services

(loose coupling by dynamic association)

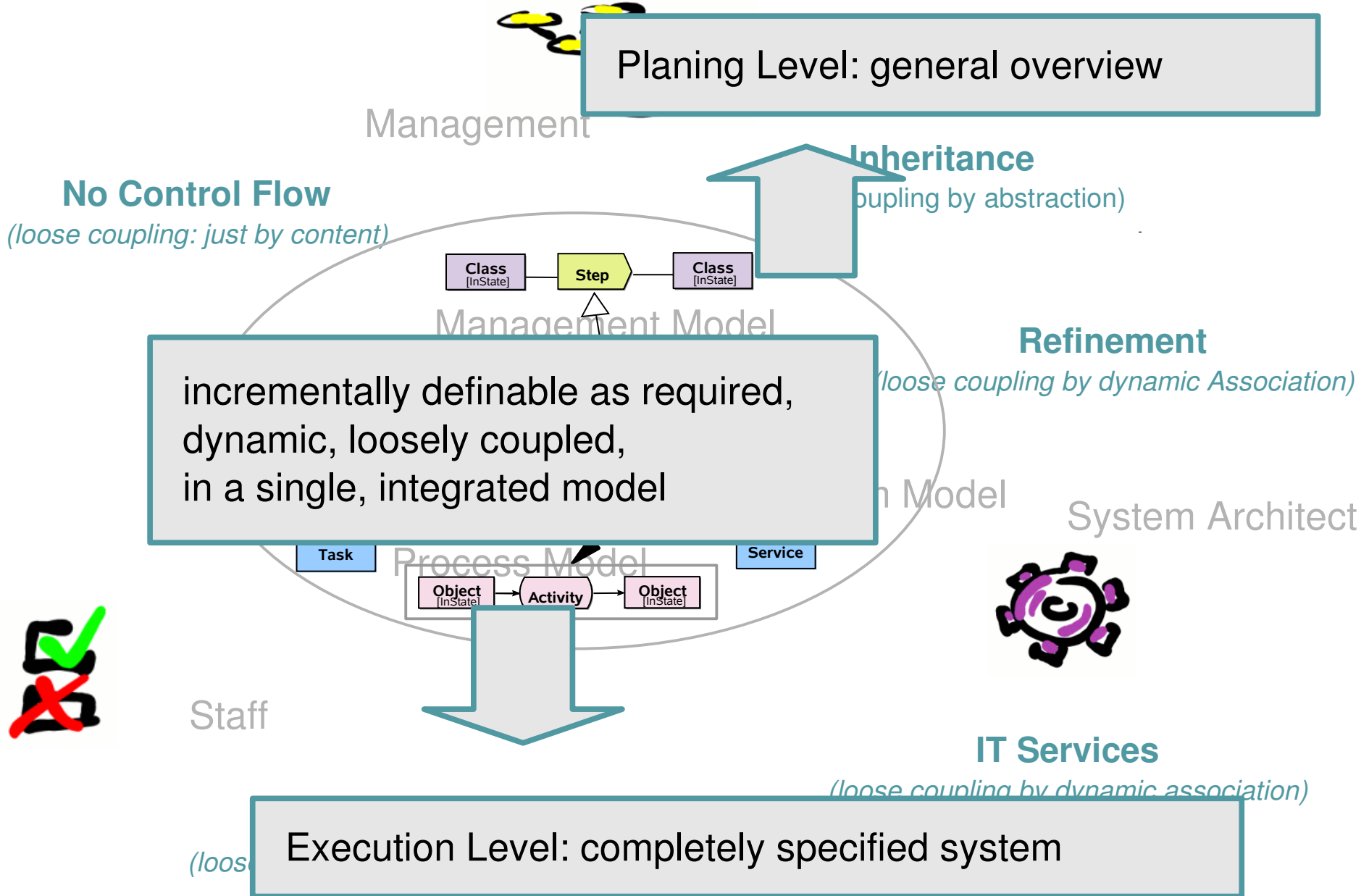
## Open for Extensions

(loose coupling by procedure-abstraction)

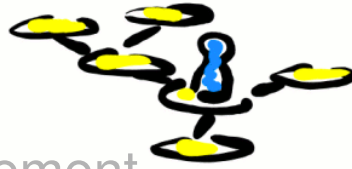


Staff

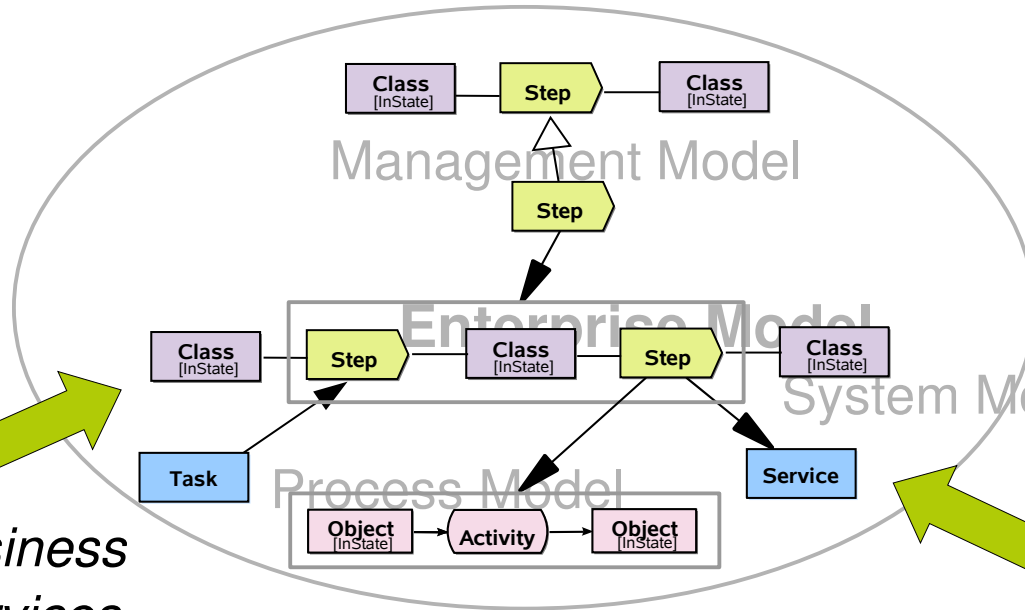
# OO-BPM Models



# SOA - Service Oriented Architecture



Management



Business Services  
Staff



IT Services



lookup  
register

Procedure (Service) Registry

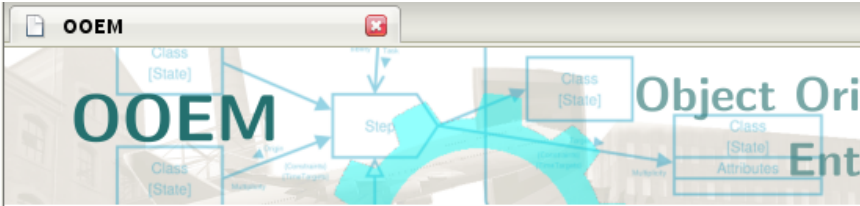
System Architect

Component

- Classes
- BPM Plans
- Procedures

# Applicability of OOBPM / OOEM

[www.ooem.org](http://www.ooem.org)



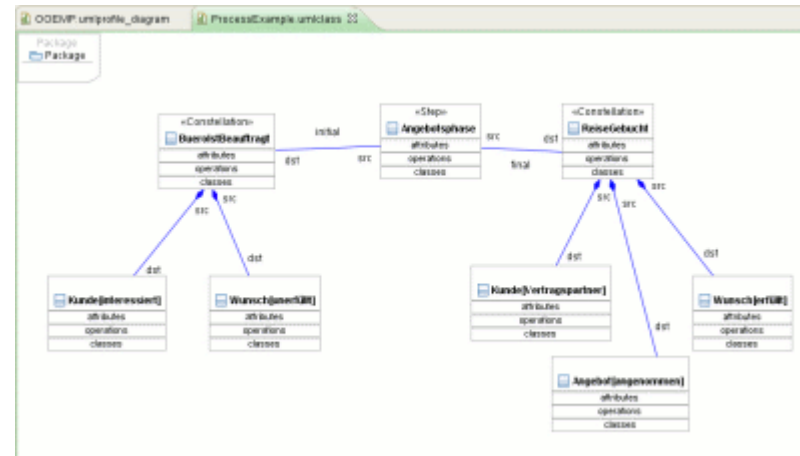
**OOEM** is a comprehensive [Enterprise Modelling Language](#).

**OOEM** Models describe the business structure and logic, as well as the processes that are executed, as well as to allow the appropriate IT systems based on the model.

More precisely, **OOEM** is based on the following technologies:

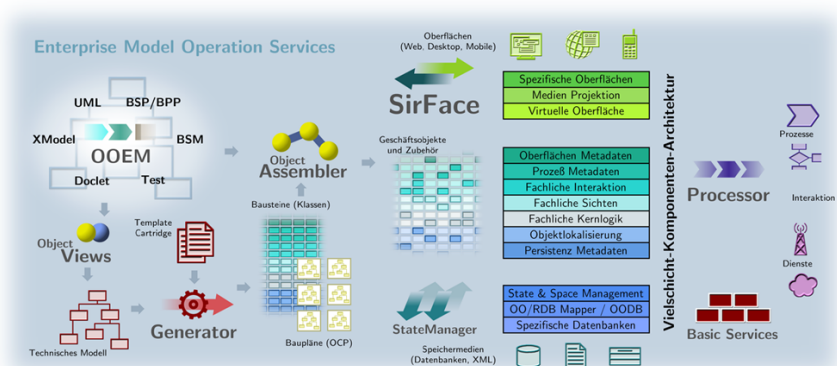
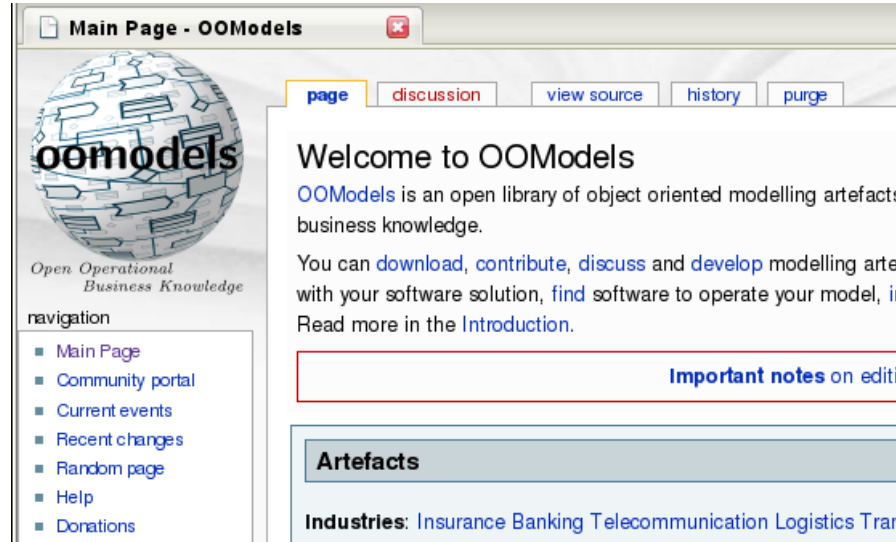
- UML
- BSP/BPP
- XModel
- BSM
- Doclet
- Test

## UML Profile for eclipse



[www.oomodels.org](http://www.oomodels.org)

## Sphenon EMOS

Welcome to OOModels

OOModels is an open library of object oriented modelling artefacts business knowledge.

You can [download](#), [contribute](#), [discuss](#) and [develop](#) modelling artefacts with your software solution, [find](#) software to operate your model, in [Read more in the Introduction](#).

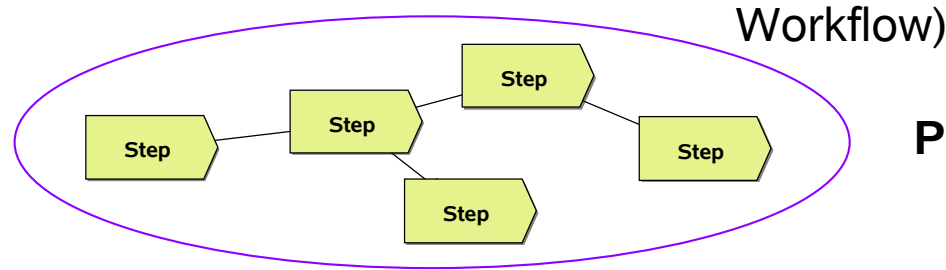
**Important notes on editing**

**Artefacts**

**Industries:** Insurance Banking Telecommunication Logistics Transportation

# Appendix

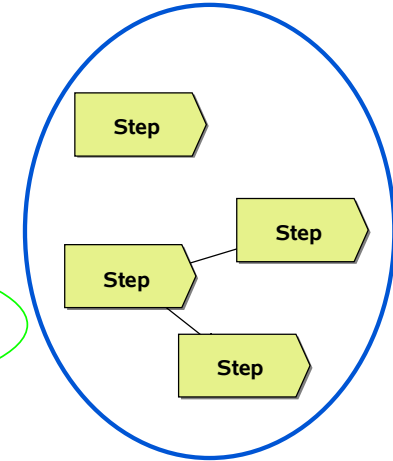
# Terms and Diagram Types



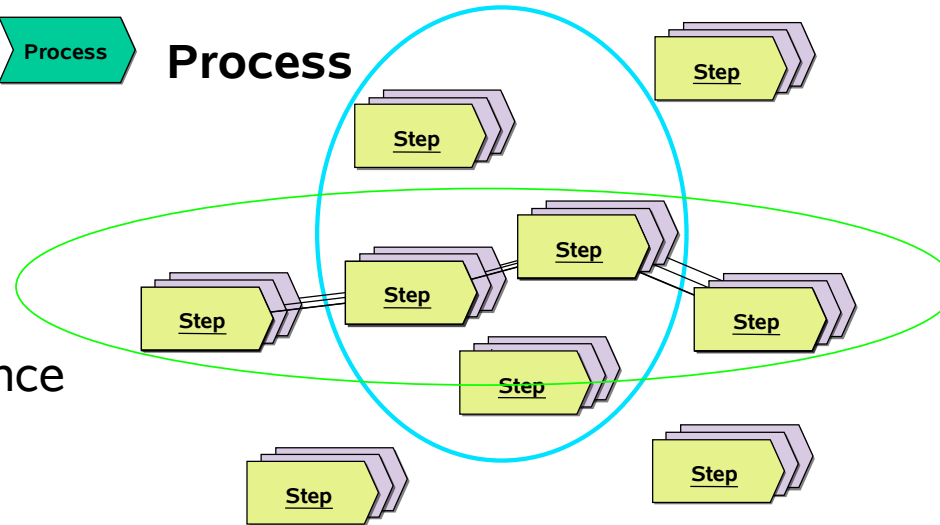
Process Stage Schema

*BSP*

Process Schema



Process

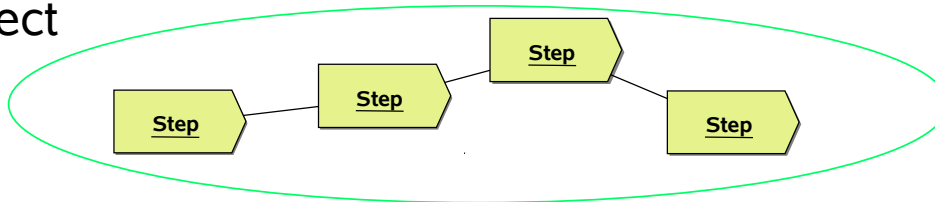


*BPP*

Workflowinstance

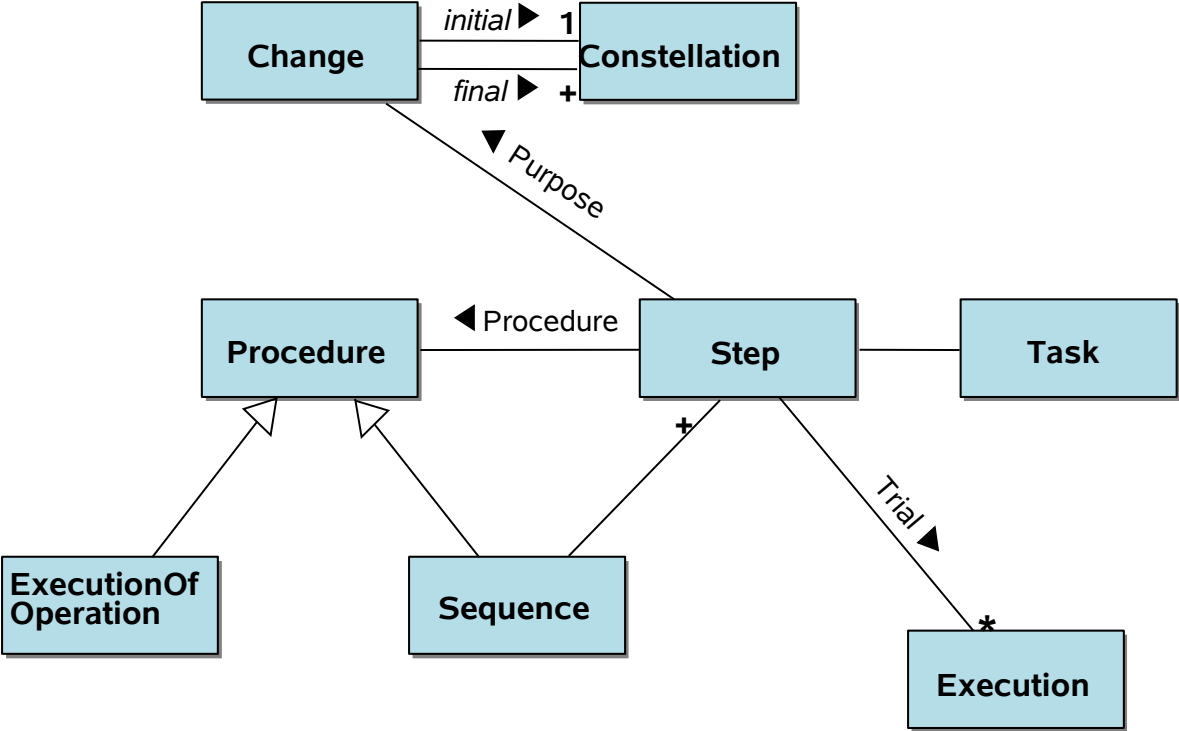
Progression Plan

Project





# Metamodel OEM (BPM)



**End**