Meta-process modeling

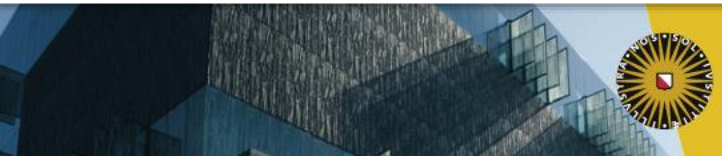
Session 4
18 February 2019

Prof.dr. Sjaak Brinkkemper
Dr. Sietse Overbeek

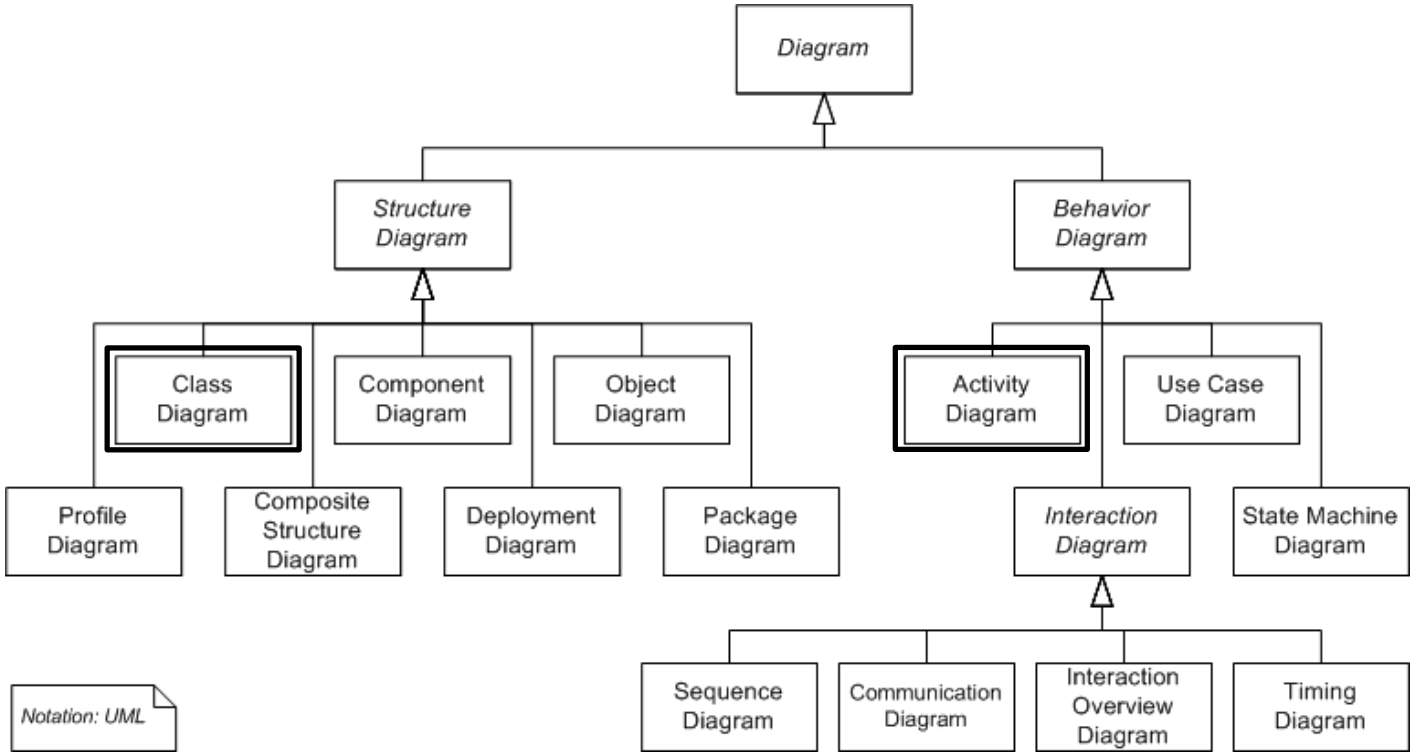
Universiteit Utrecht

Agenda

- Data modeling assignment
- Introduction to meta-process modeling
- PDD notation
- Meta-modeling assignment

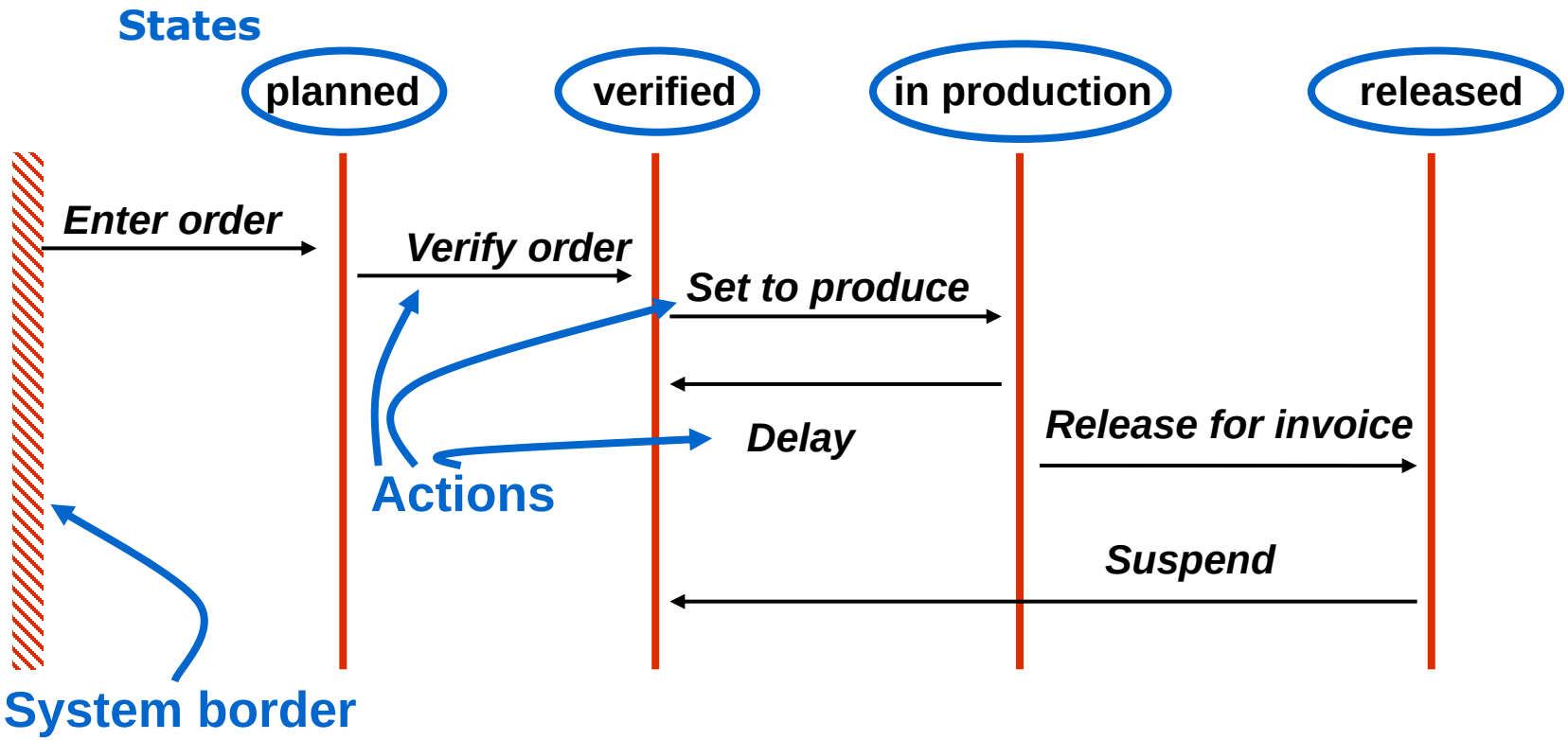


Overview of the UML 'diagram stack' (or 'language' stack)

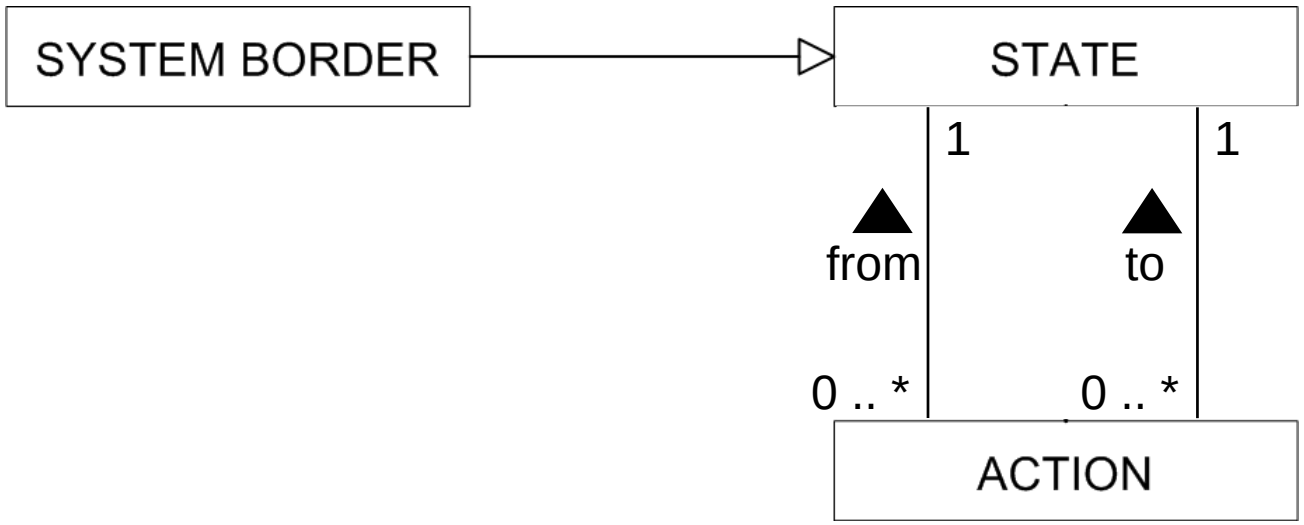


Assignment: sequence diagram

Create the meta-model for the Sequence diagram

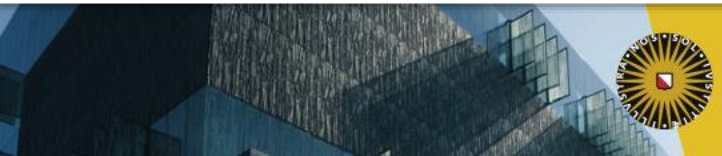


Assignment: sequence diagram



Agenda

- Data modeling assignment
- Introduction to meta-process modeling
- PDD notation
- Meta-modeling assignment

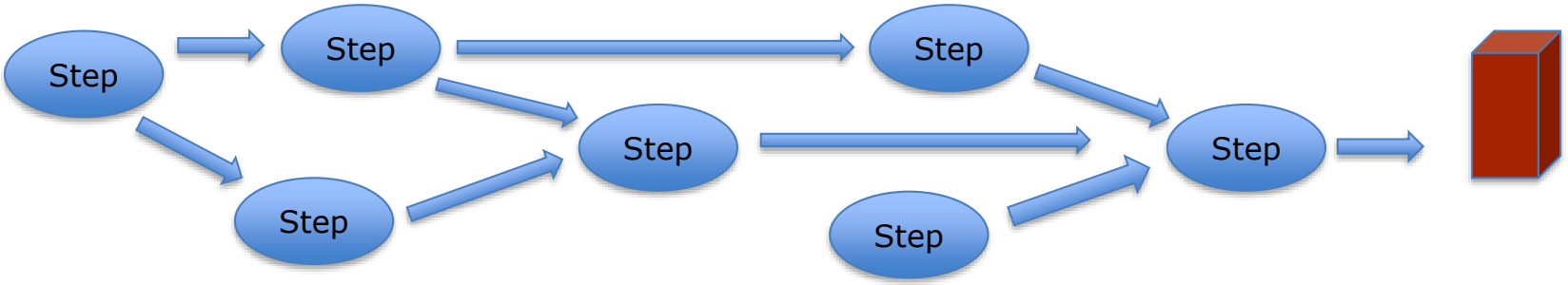


Process

Definition

A set of partially ordered steps intended to produce a desired product

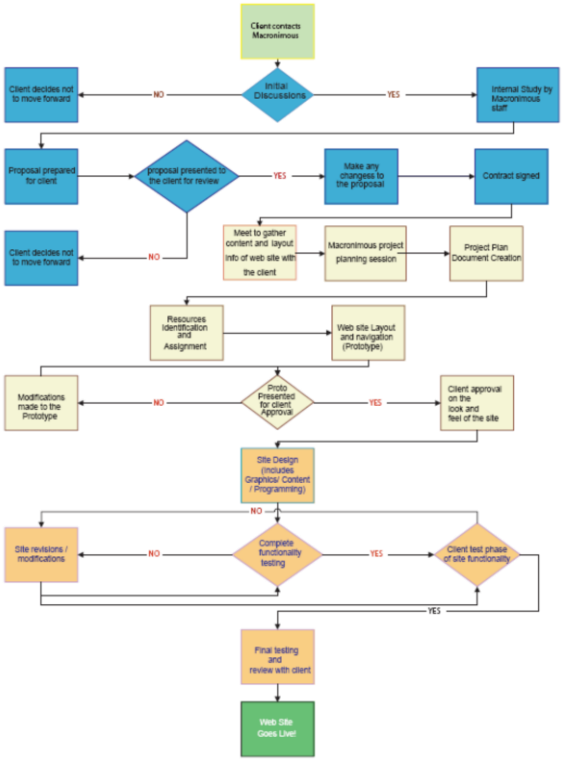
Heineman, Botsford, Caldiera, Kaiser, Kellner, and Madhavji (1994)

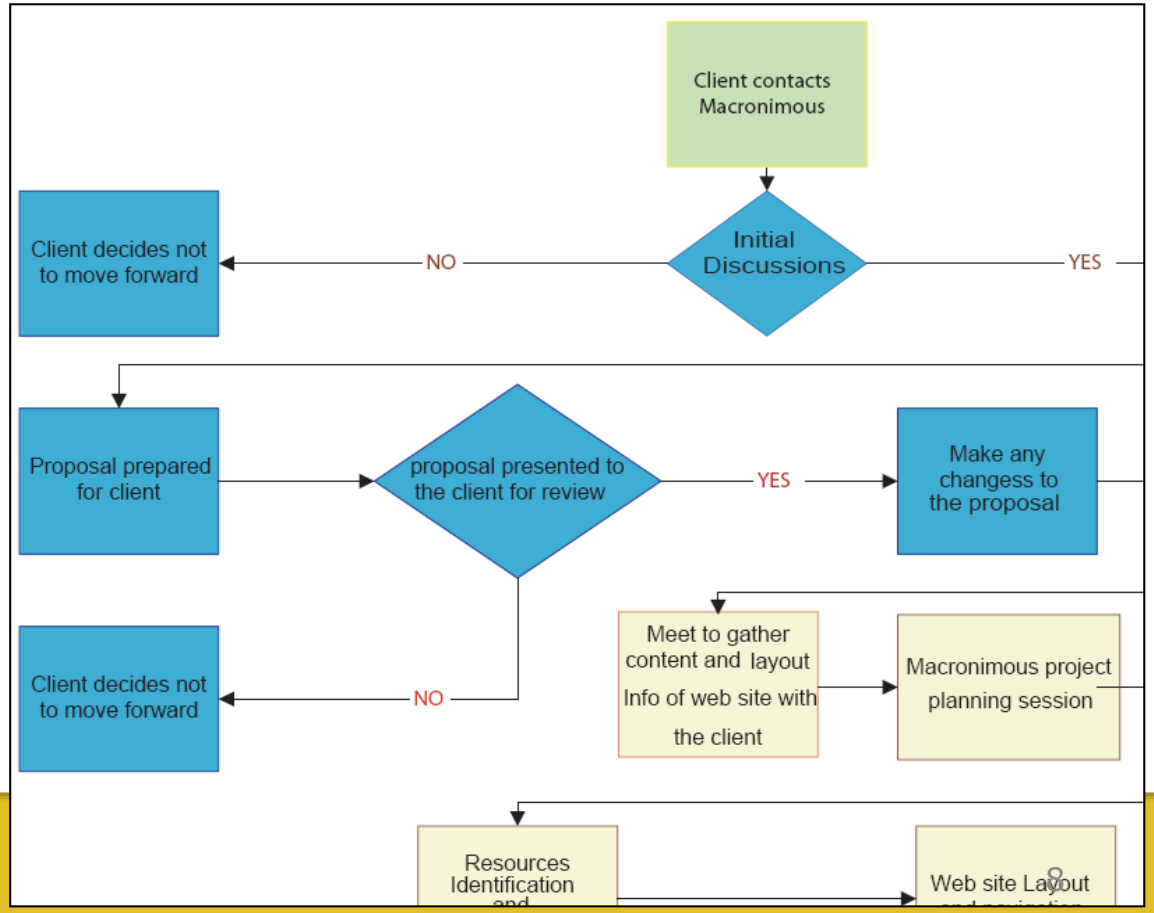


(partially ordered = any two steps are not required to be related to each other)

Processes (1)

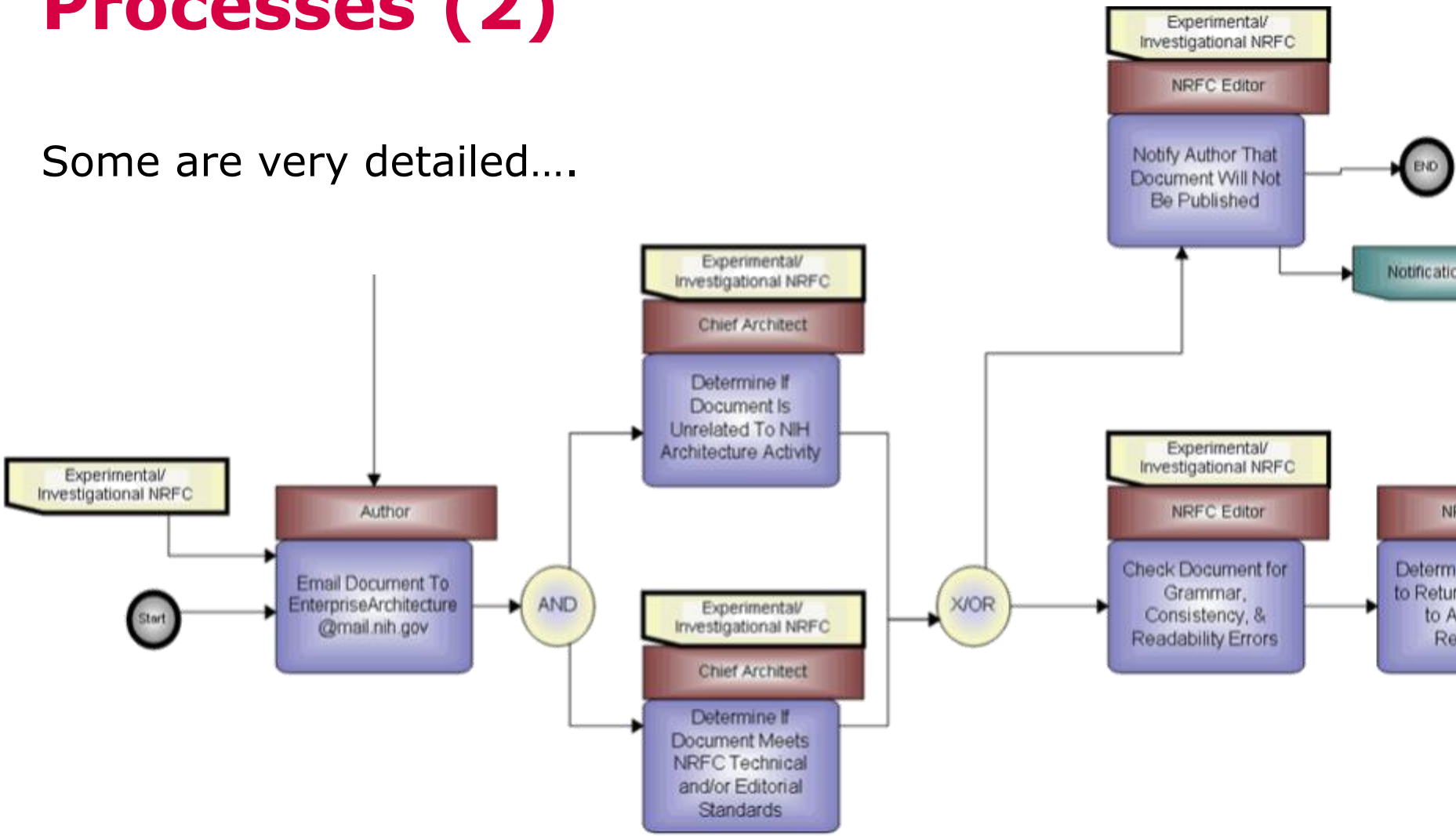
- Process models are widely documented for various purposes





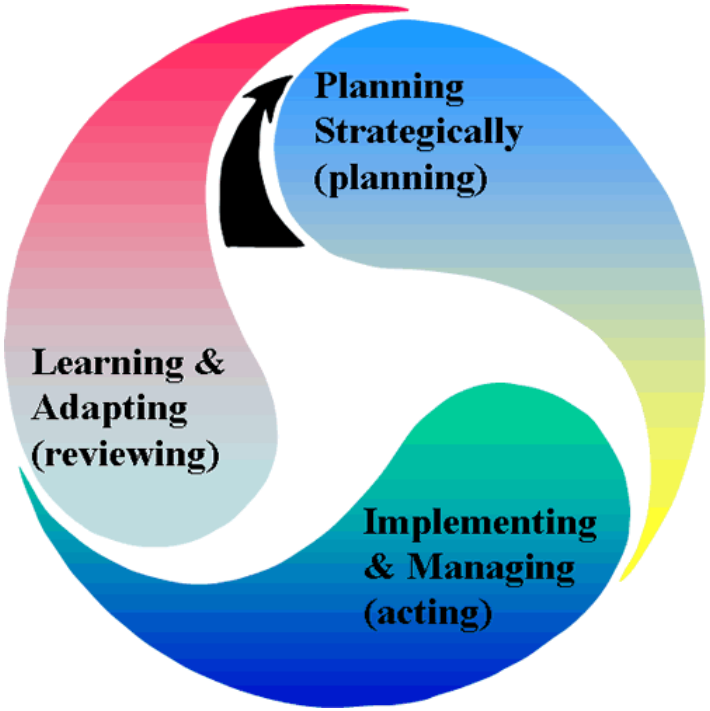
Processes (2)

Some are very detailed….

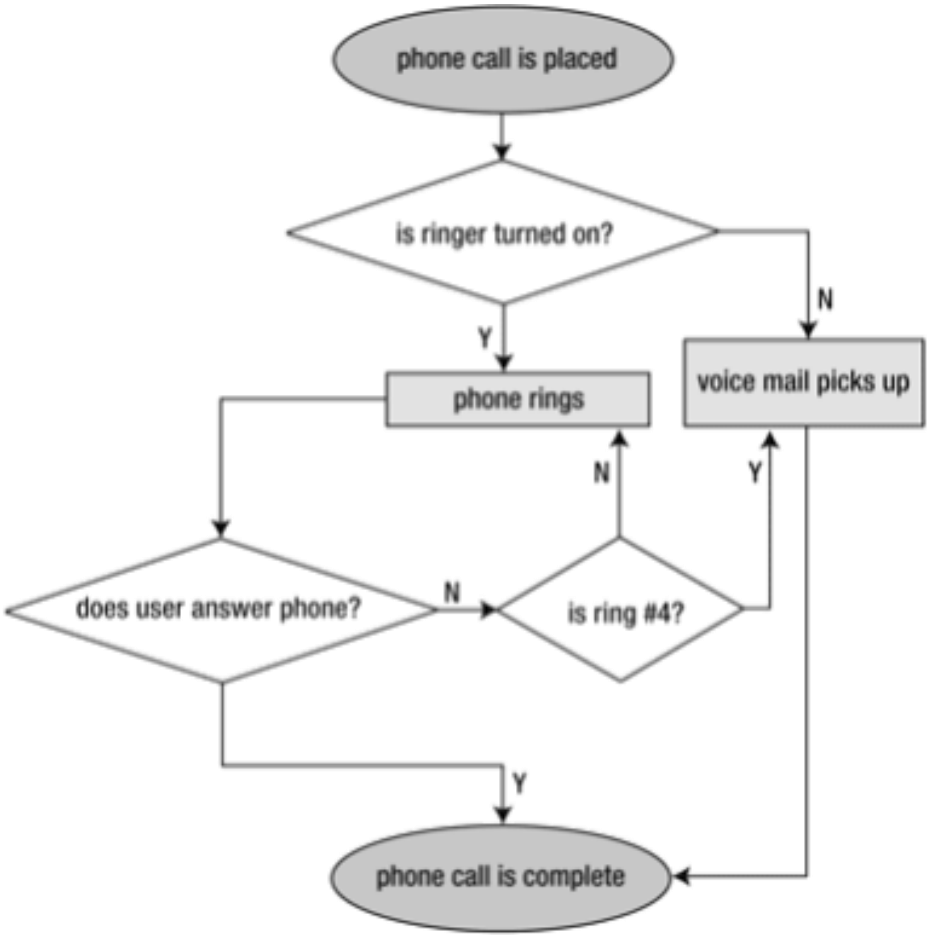


Processes (3)

…while others are not



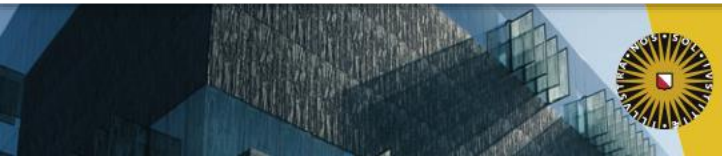
Process concepts



- Activity
- Decision
- Condition
- Step
- Action
- Flow
- Trigger
- State
- Repetition
- Sub-process
- Termination
- Etc.

Difficulty of processes

- Many terms for process perspective: process, action, activity, step, procedure, practice, task, phase, stage, plan
- No formal abstraction levels of the process perspective
- Process size is difficult to measure
- Processes of different granularity levels are easily mixed
- It's the result that counts, i.e. the process is subordinate to the deliverable

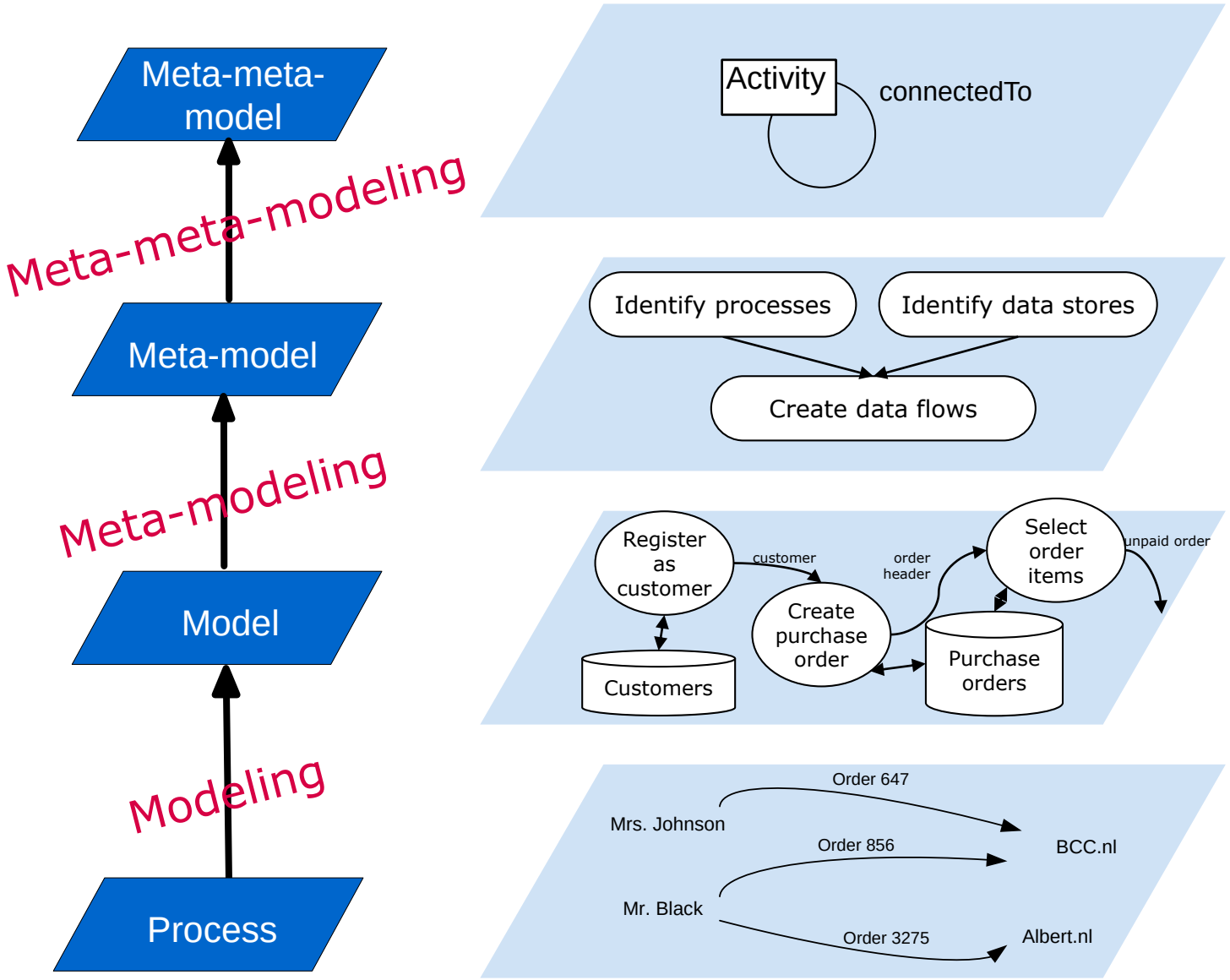


Abstraction of concepts

Product perspective	↔	Process perspective
System	↔	Project
Deliverable, Milestone document	↔	Phase
Document, Product	↔	Activity
Concept	↔	Step

Abstraction levels in the product perspective seem natural, whereas those in the process perspective are artificial

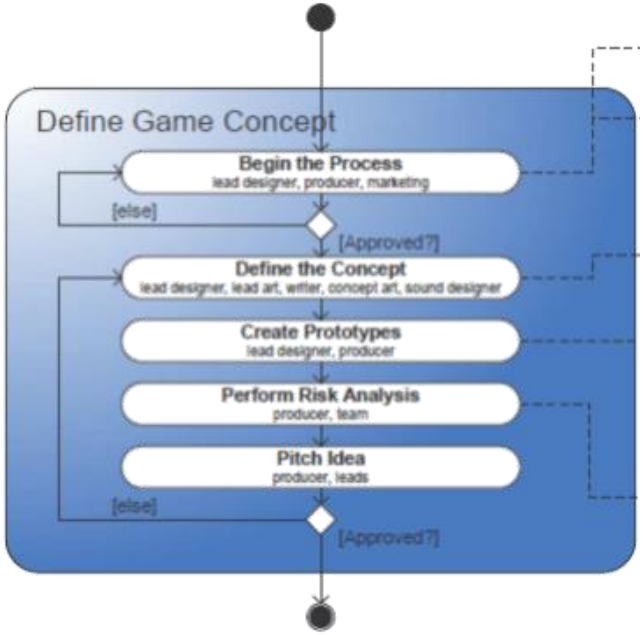
DFD-example


Meta-meta-model
Meta-meta-modeling
Meta-model
Meta-modeling
Model
Modeling
Process
Activity
connectedTo
Identify processes
Identify data stores
Create data flows
Register as customer
customer
order header
Select order items
unpaid order
Create purchase order
Customers
Purchase orders
Order 647
Mrs. Johnson
BCC.nl
Order 856
Mr. Black
Order 3275
Albert.nl

Meta-process modeling

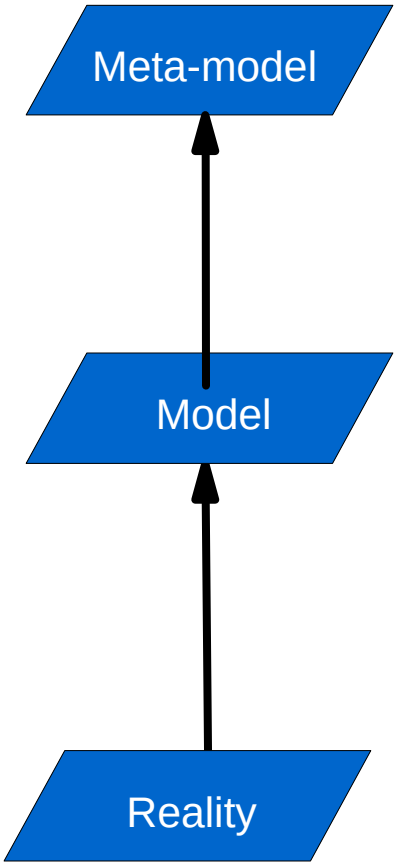
- A meta-process model is a model of a methodical process

- Examples:
  - Development project plan
  - Design process
  - Process improvement process
  - Daily build process



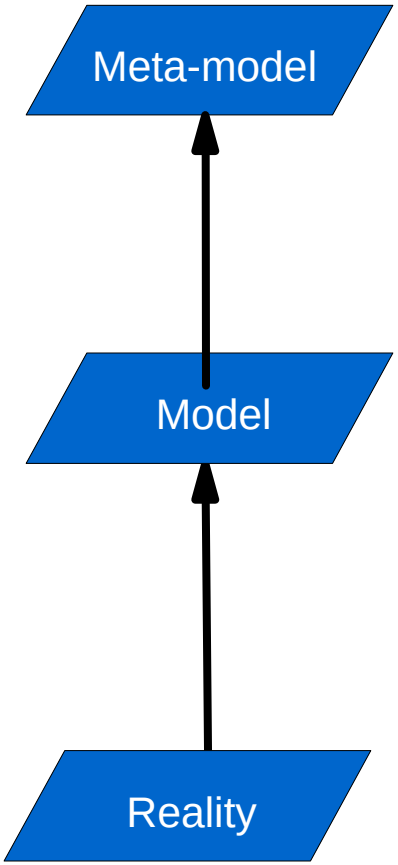
- In the PDD notation, meta-process models are modeled with a similar notation as the UML activity diagrams

Model vs. meta-model for diagrammatic techniques

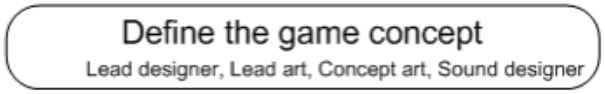


- Method knowledge
  Identify processes, Create data flows
- Domain knowledge
  Register as customer, unpaid order
- Occurence knowledge
  “Mrs. Johnson places order 856”

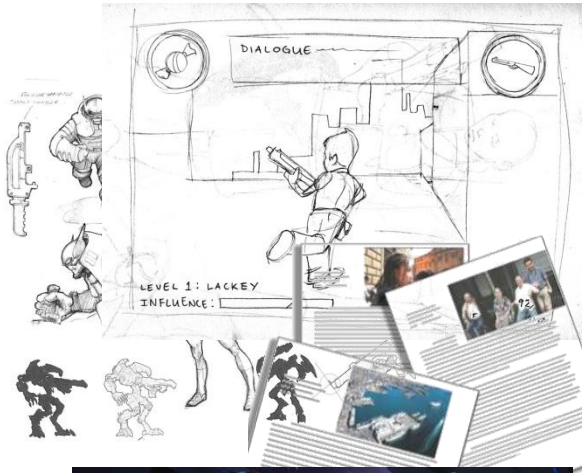
Model vs. meta-model for methods



- Method knowledge
  "Lead designer, lead art, writer, concept art and sound designer define a game concept"



- Domain knowledge
  "Create game concept for Mass Effect 2"



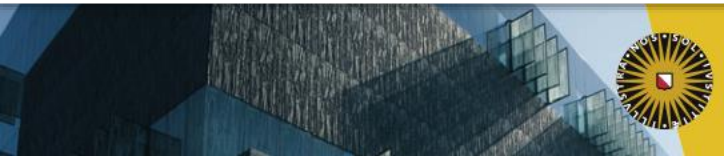
- Occurence knowledge
  "Play Mass Effect 2"

Why the (slightly) different perspective?

- M0 is reality layer (Instance)
- M1 is development layer (Model)
- M2 is method layer (Meta-model)

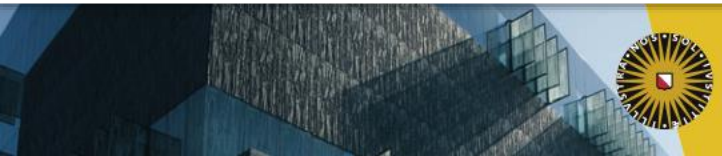
- There is still some discussion concerning the model vs. meta-modeling distinction, but for the purpose of being able to model information systems methods, we choose to follow the above structure

- Further reading, e.g. Kurtev (2007)



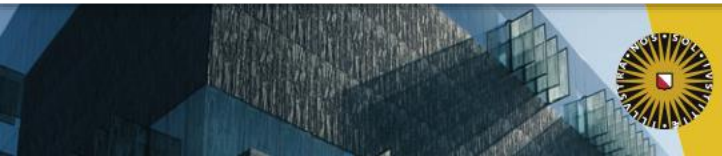
Usage of (meta-)process modeling

- Documenting the process
- Communicating the process
- Comparing processes
- Re-using the process
- Development of tool support



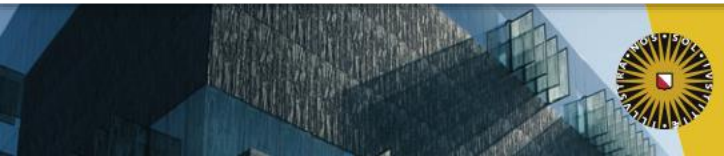
Agenda

- Data modeling assignment
- Introduction to meta-process modeling
- PDD notation
- Meta-modeling assignment

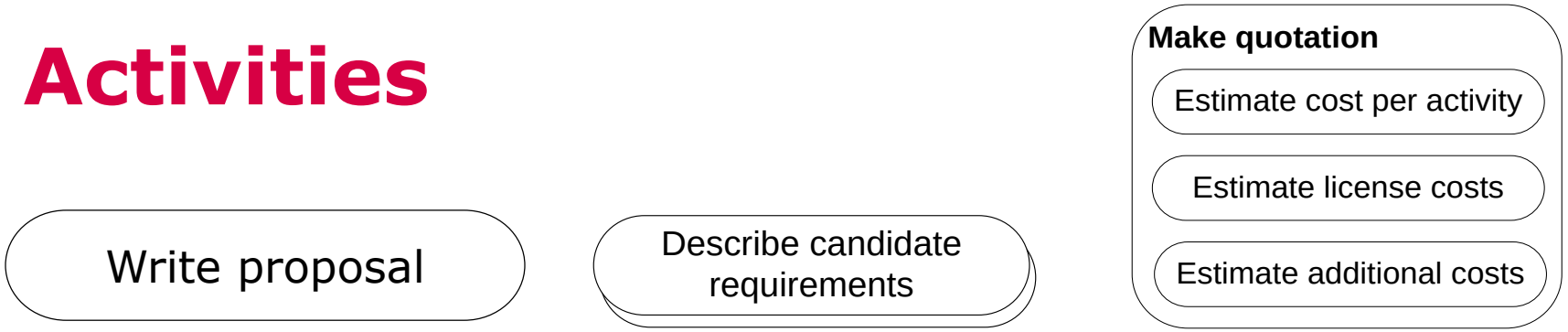


Notation

- Activities
- Types of activities
- Activities
  - sequential
  - unordered
  - concurrent
  - conditional
- Roles
- Levels

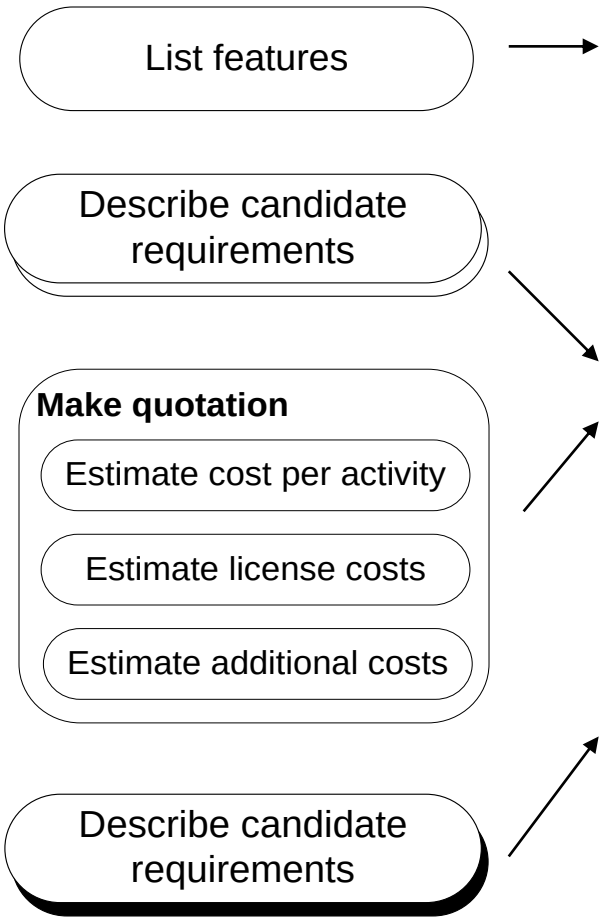


Activities



- Activities are modeled in boxes with round corners
- Naming convention: At the ***lowest*** level the activity names all start with a verb and followed by an objects description
  - e.g. Identify users, Develop system interaction, Estimate licence costs, Describe requirements.
- At ***higher*** levels this is recommended, but not enforced.
  - Phase names can be nouns, e.g. Requirements, Implementation (see also levels of meta-processes)

Types of activities

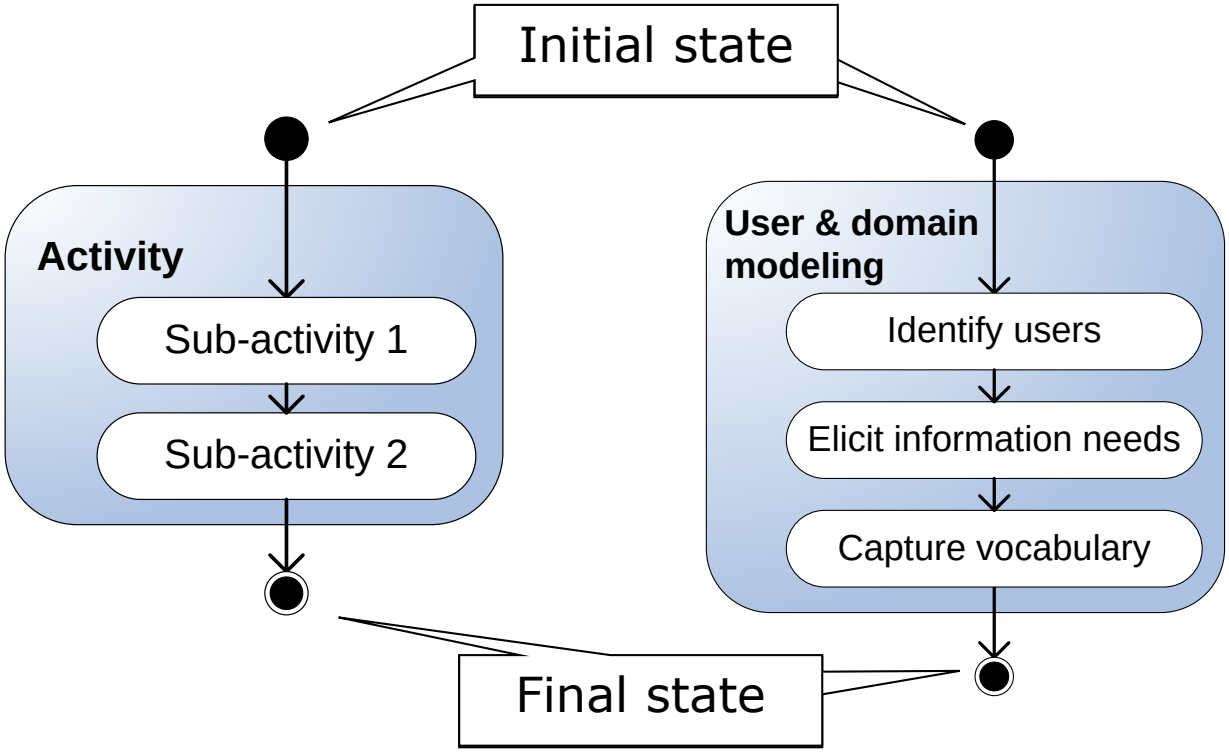


Standard Activity: activity without sub-activities (atomic activity)

Complex activity: with sub-activities

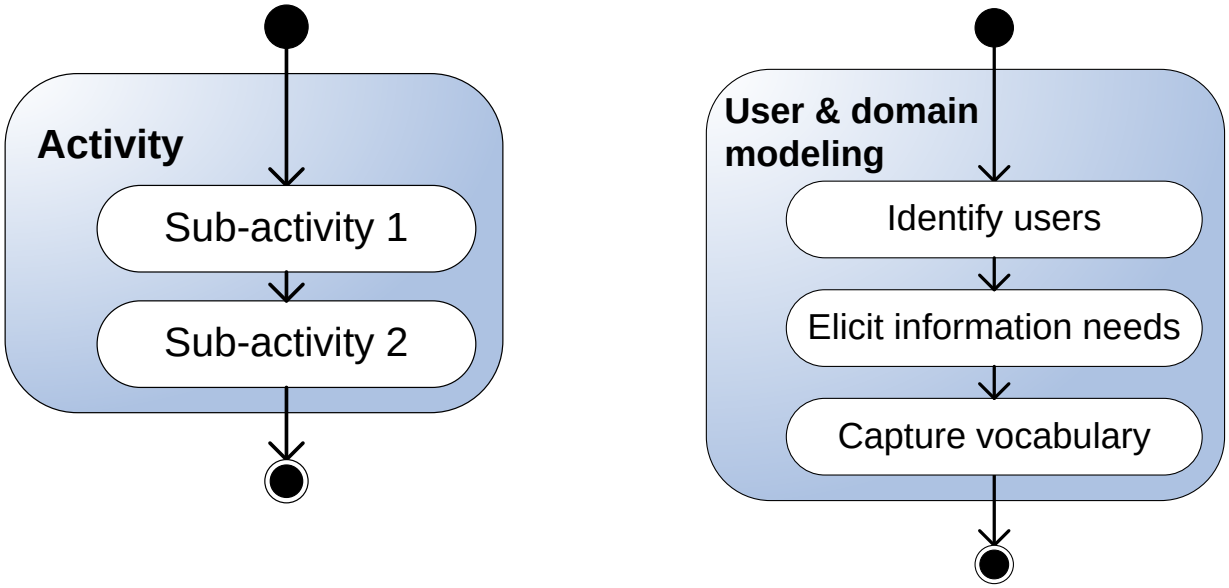
- Open activity: subactivities are *known in this context*, i.e. the meta-model details can be found in this or another meta-model of the method engineering study at hand.
- Closed activity: subactivities are *not known in this context*

Initial and Final state



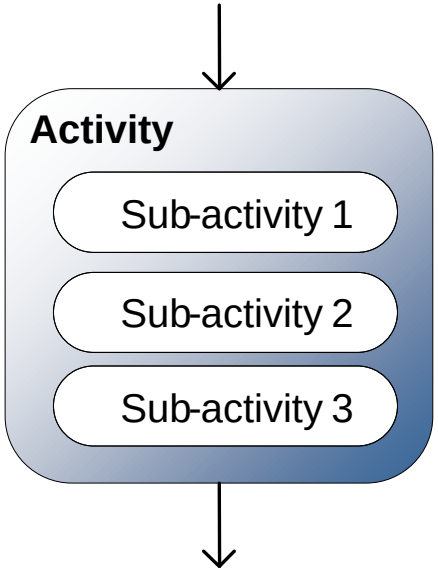
- Initial state is before any activity is started
- Final state is after the end of *all* activities
- There is always just one Initial state and one Final state. (sometimes multiple final states to avoid long arrows)

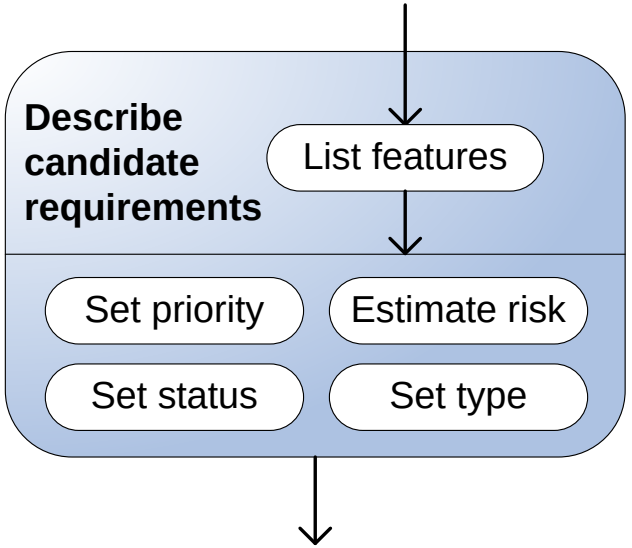
Sequential activities



- Sequential activities are activities that need to be carried out in a *pre-defined order*.

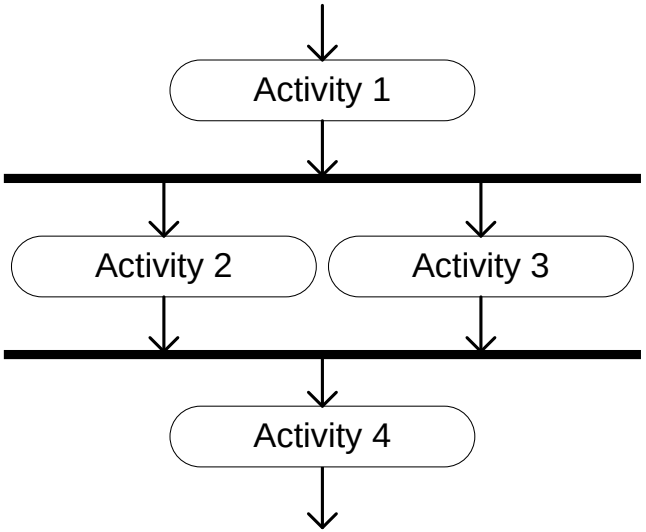
Unordered activities

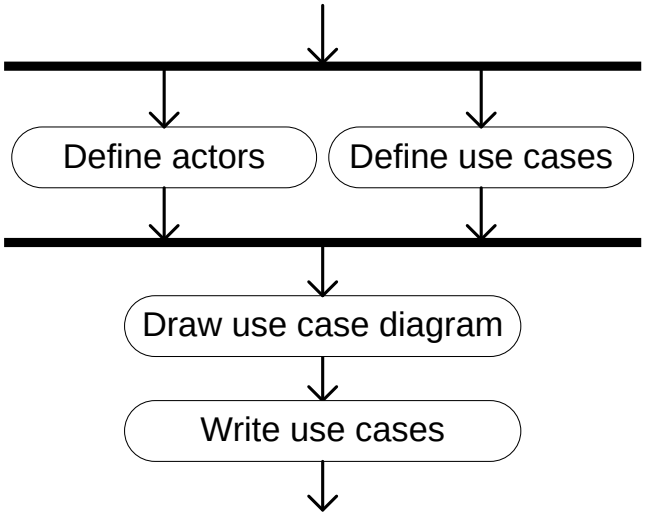




Unordered activities are used when sub-activities of an activity can be *executed in any order*.

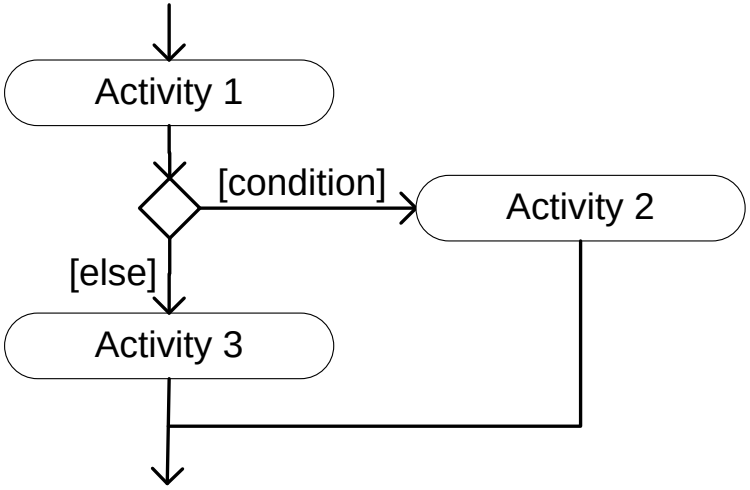
Concurrent activities

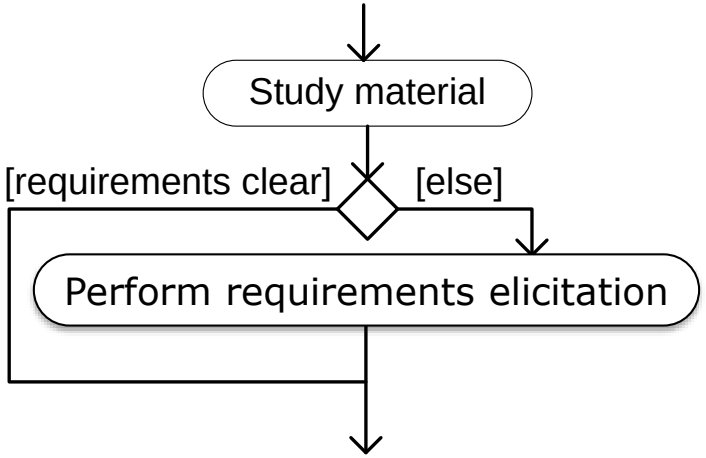




Concurrent activities are executed in parallel, but upon completion synchronization is needed before continuing.

Conditional activities (1)



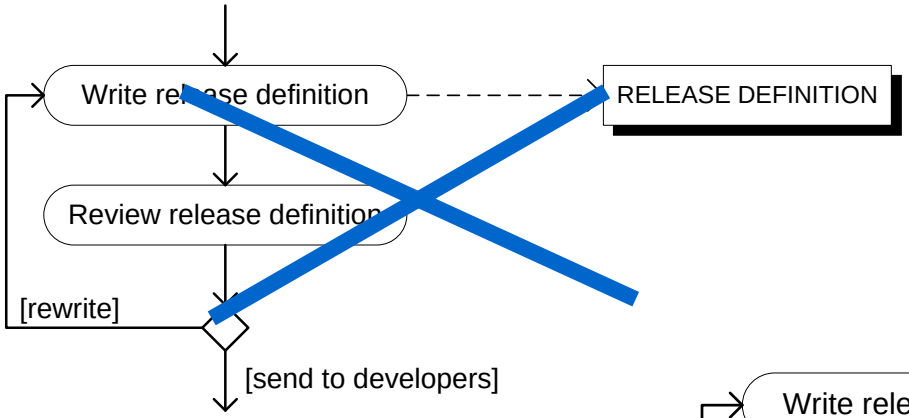


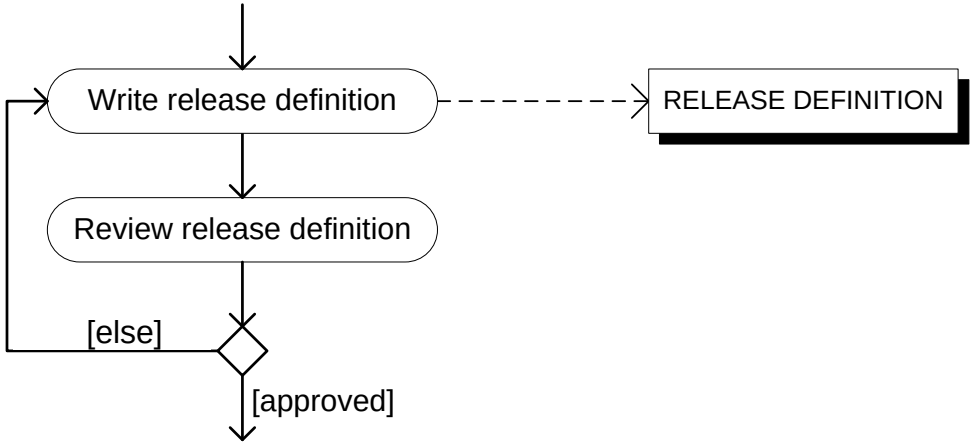
Conditional activities are activities that are only carried out if a pre-defined condition is met.

The condition is stated as a Boolean expression between square brackets [ ] .

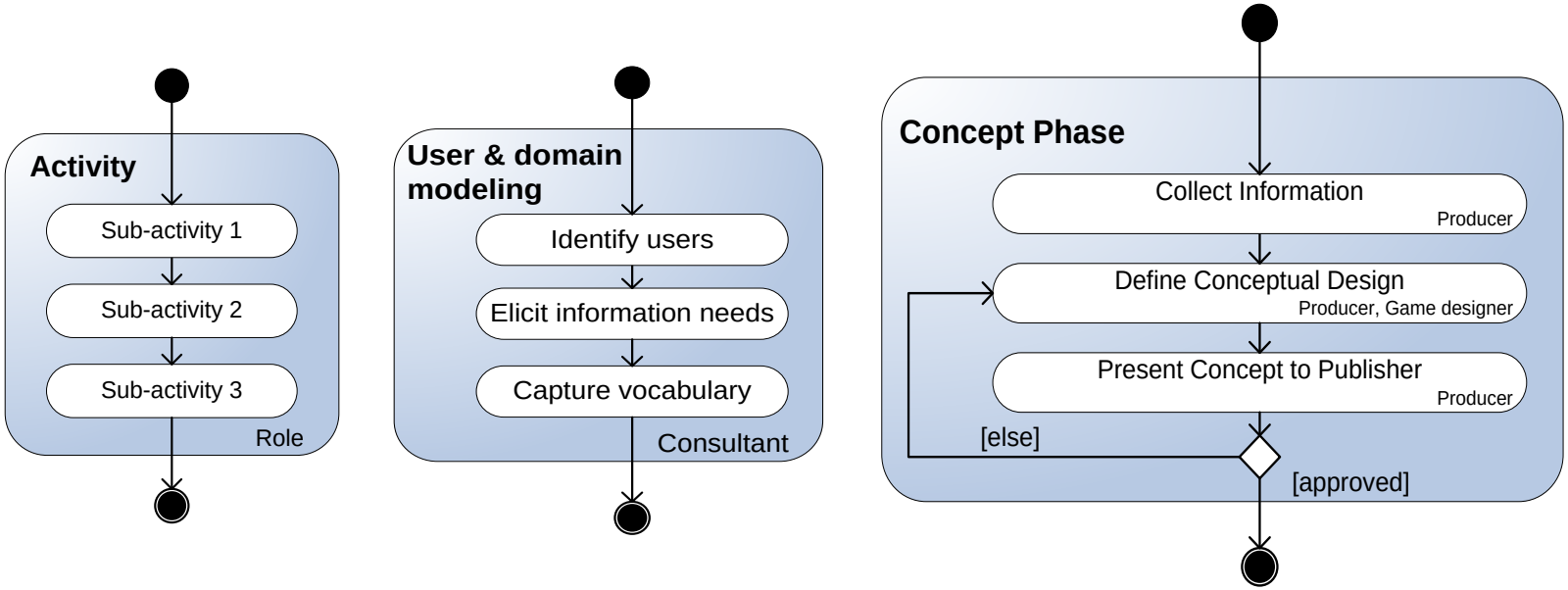
Conditional activities (2)

What's wrong in this figure?





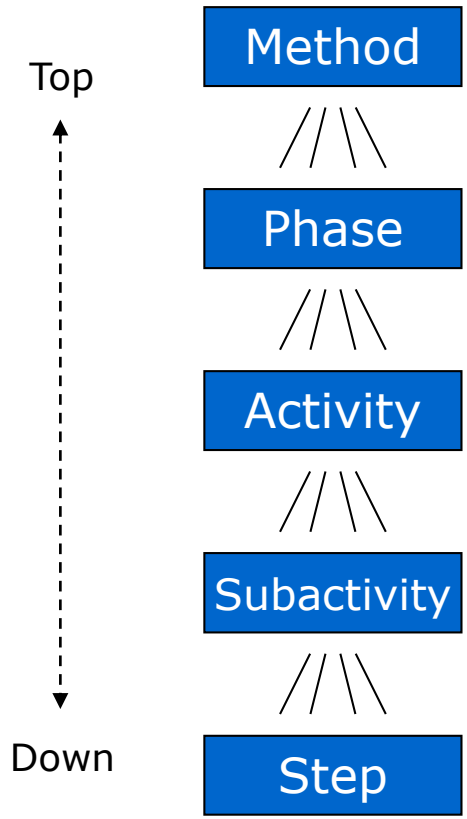
Roles



In some methods it is explicitly stated by which individuals or organizational role the process is to be carried out.

In this case, the role is indicated in the activity. If necessary, roles can also be indicated in sub-activities.

Meta-model leveling



Needs

Overview:
Decision making, time, budget

Detail:
Instructions, execution dependencies

Stakeholders

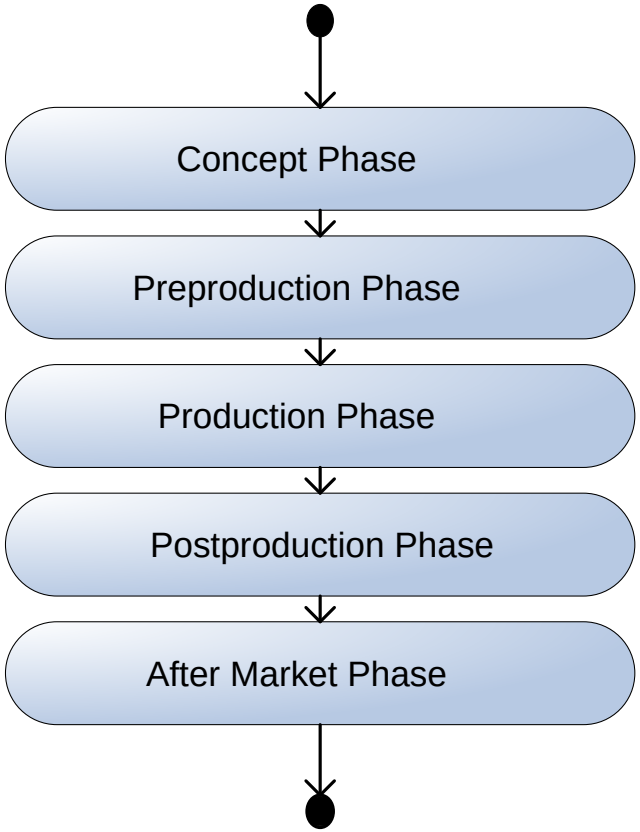
Project manager, general management

End-user, Software engineer, architect, analist

Levels of meta-process models (1)

A. High-level meta-process models
   - Hierarchical models: show the detailing from high to low level activities
   - Staged models: show the sequence of major phases in activities

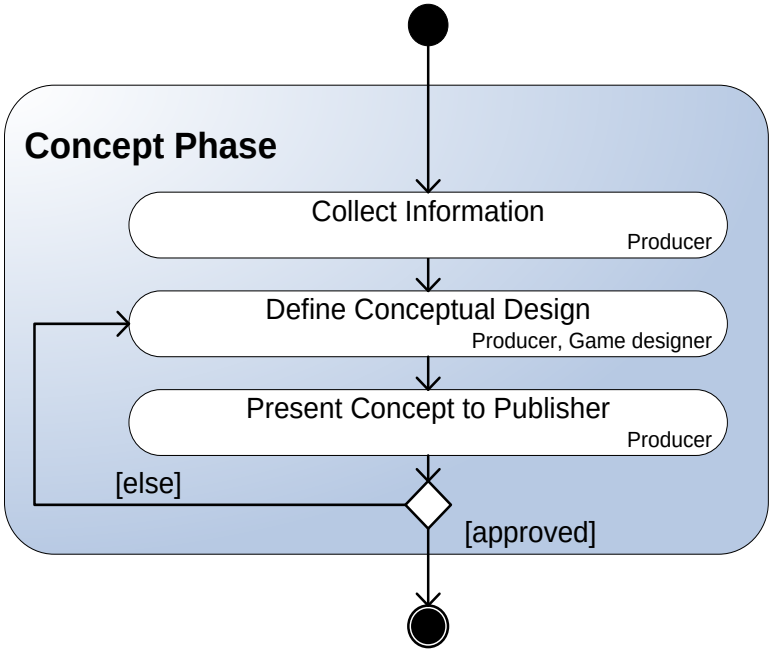
Note: activity names can be nouns



Rabin, S. (2006). *Introduction to Game Development.* Hingham, MA: Charles River Media.

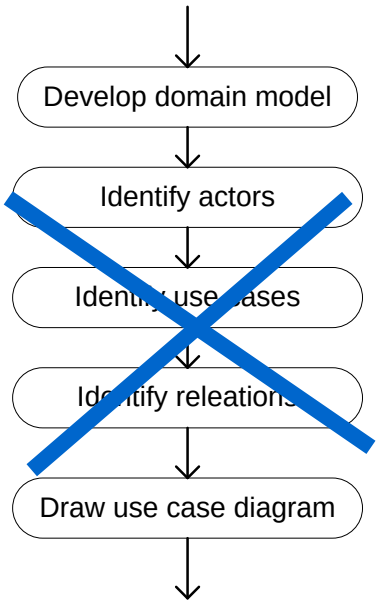
Levels of meta-process models (2)

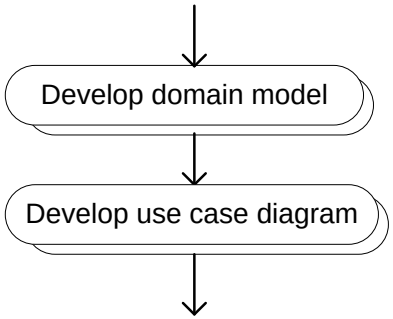
B. Low-level meta-process models
- Leveling of activities
- *Data-flow* models: the data flows
- *Control-flow* models: triggers on flows

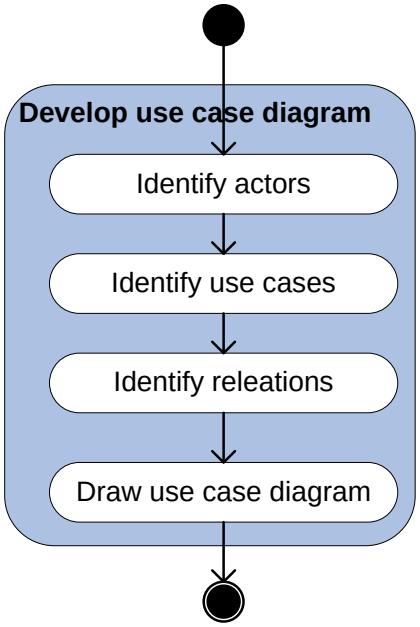


Finding the balance

- What's wrong in this figure?

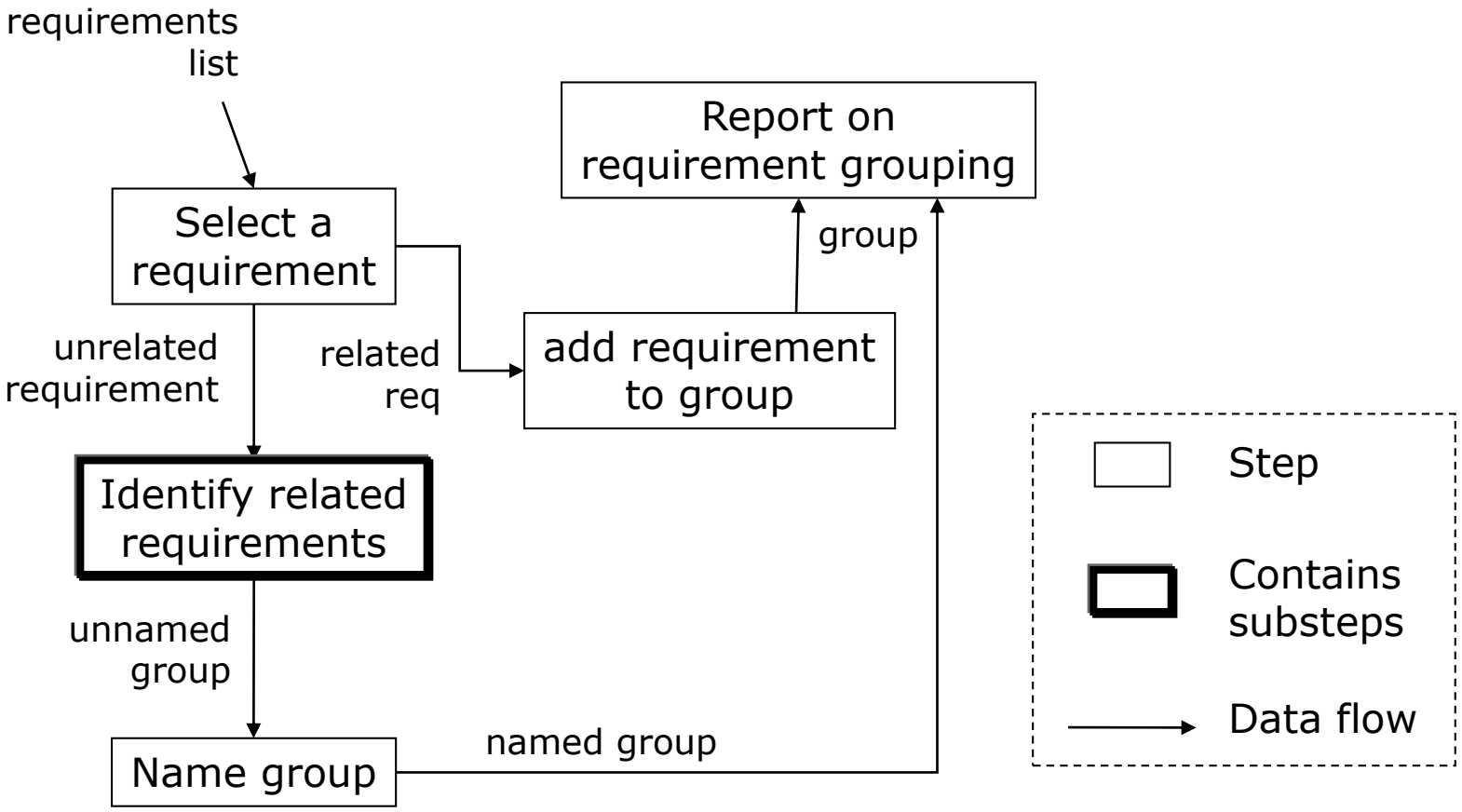






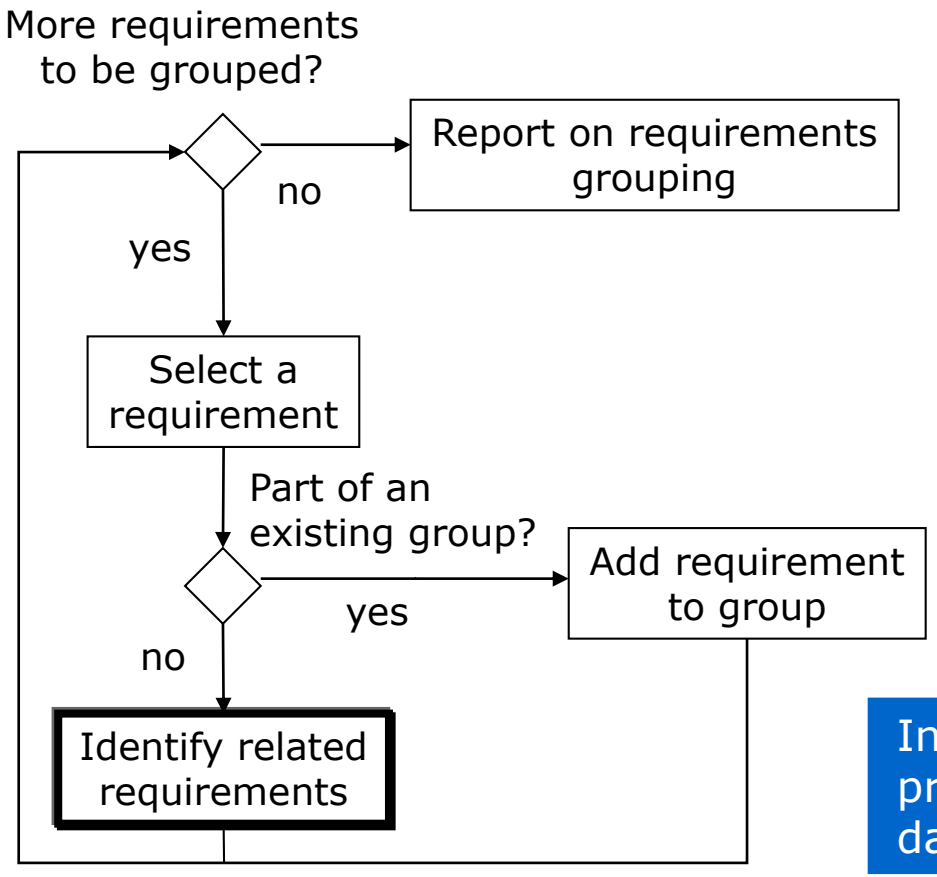
Activities in one diagram should be at the same abstraction level

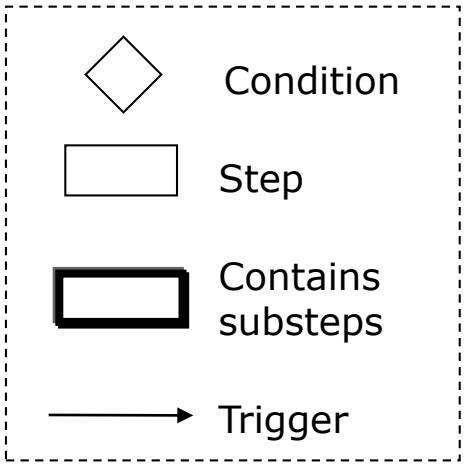
Data flow model



Data goes from one step to the other

Control flow model





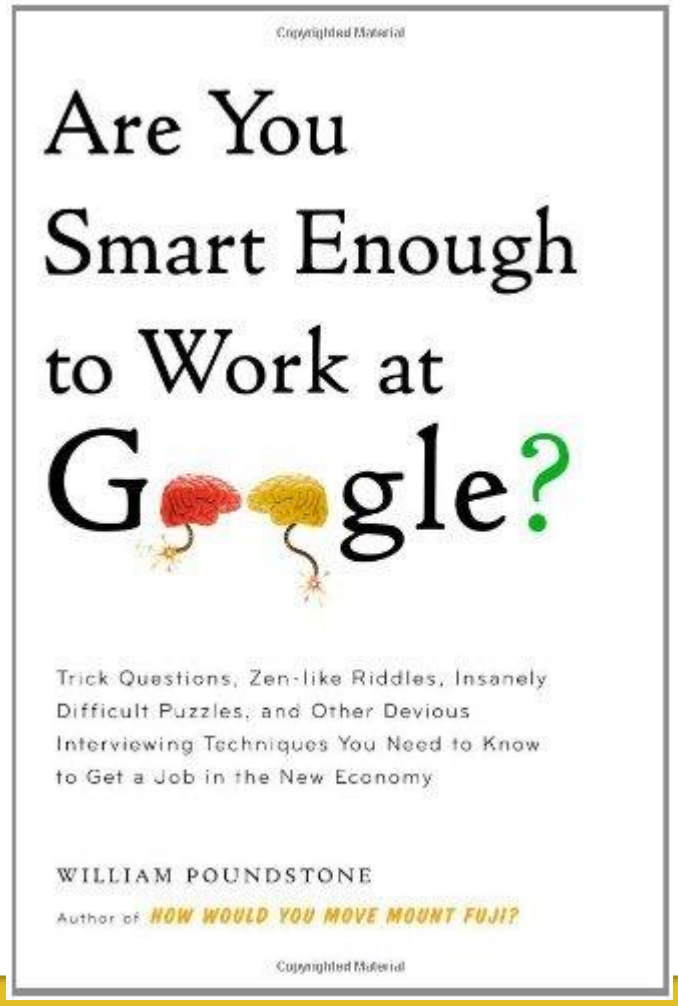
In a meta-process model we prefer control flows, as the data is in the meta-data model.

Control is transfered from one step to the other

Break

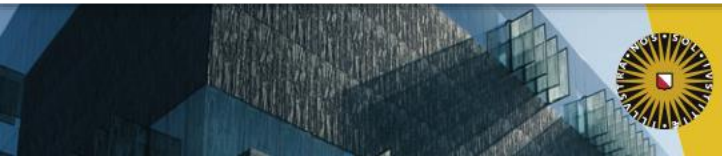
You get on a ski lift at the bottom of a mountain and take it all the way up to the top.

What fraction of the lift's chairs do you pass?



Wrap-up

- Activities
- Types of activities
- Activities
  - sequential
  - unordered
  - concurrent
  - conditional
- Roles
- Levels



Process-deliverable diagram

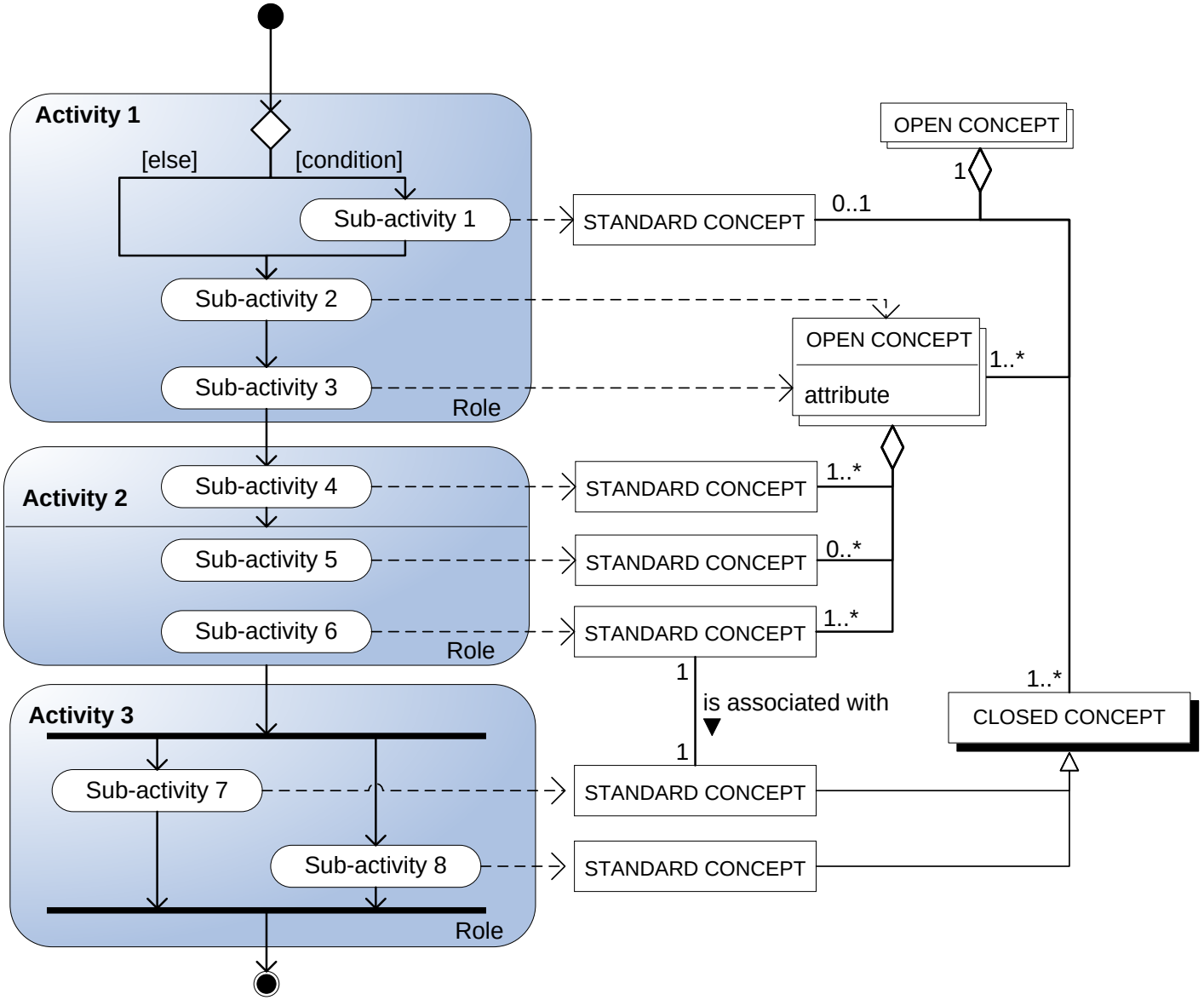
- The meta-data model and meta-process model are connected in the so-called Process-Deliverable Diagram, abbreviated PDD.

- Arrows link the activity to the deliverable it has produced

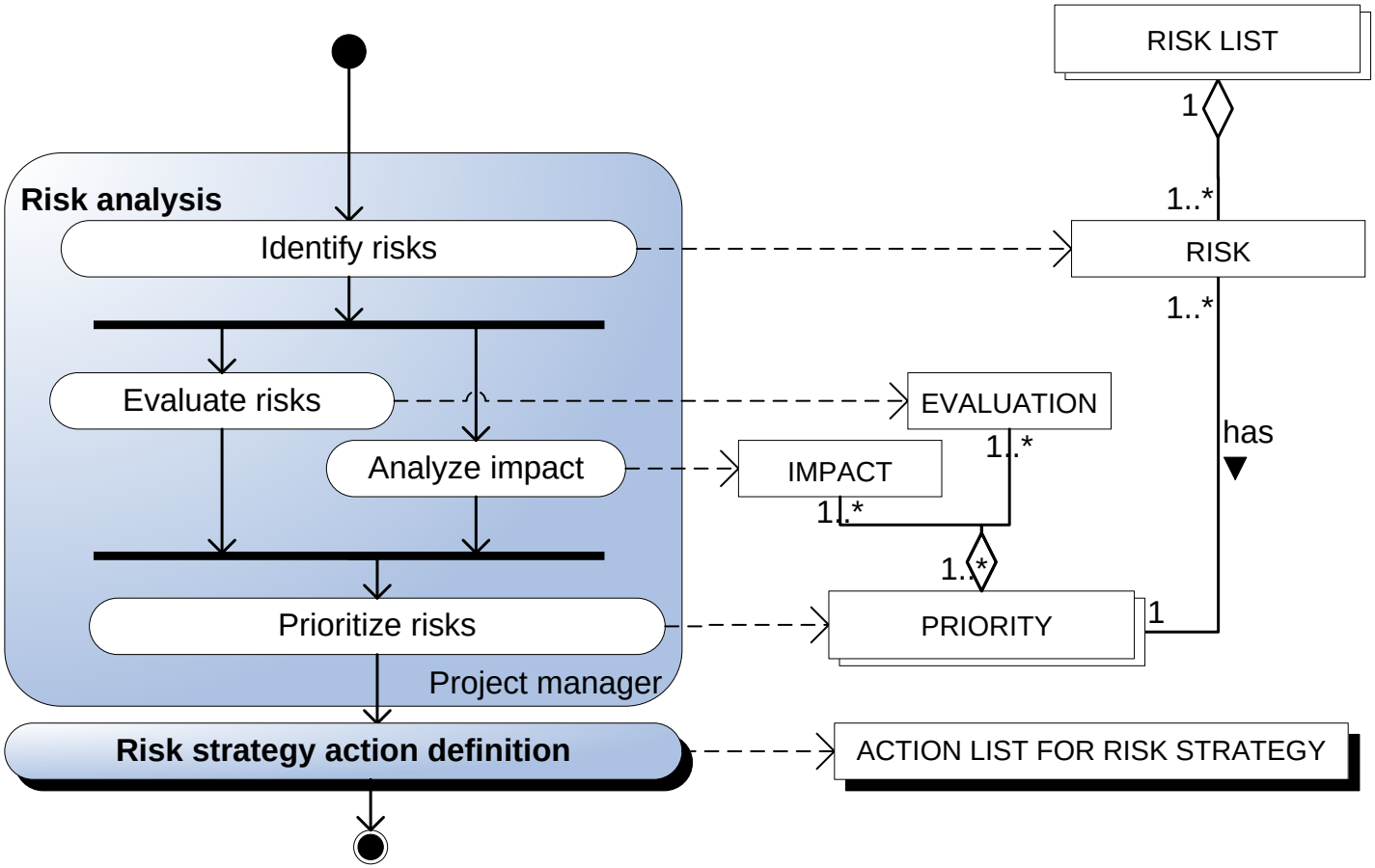
  ---------->

- Input arrows, e.g. which concepts are required an activity are NOT shown, as all produced deliverables are assumed to be generally available

Generic process-deliverable diagram



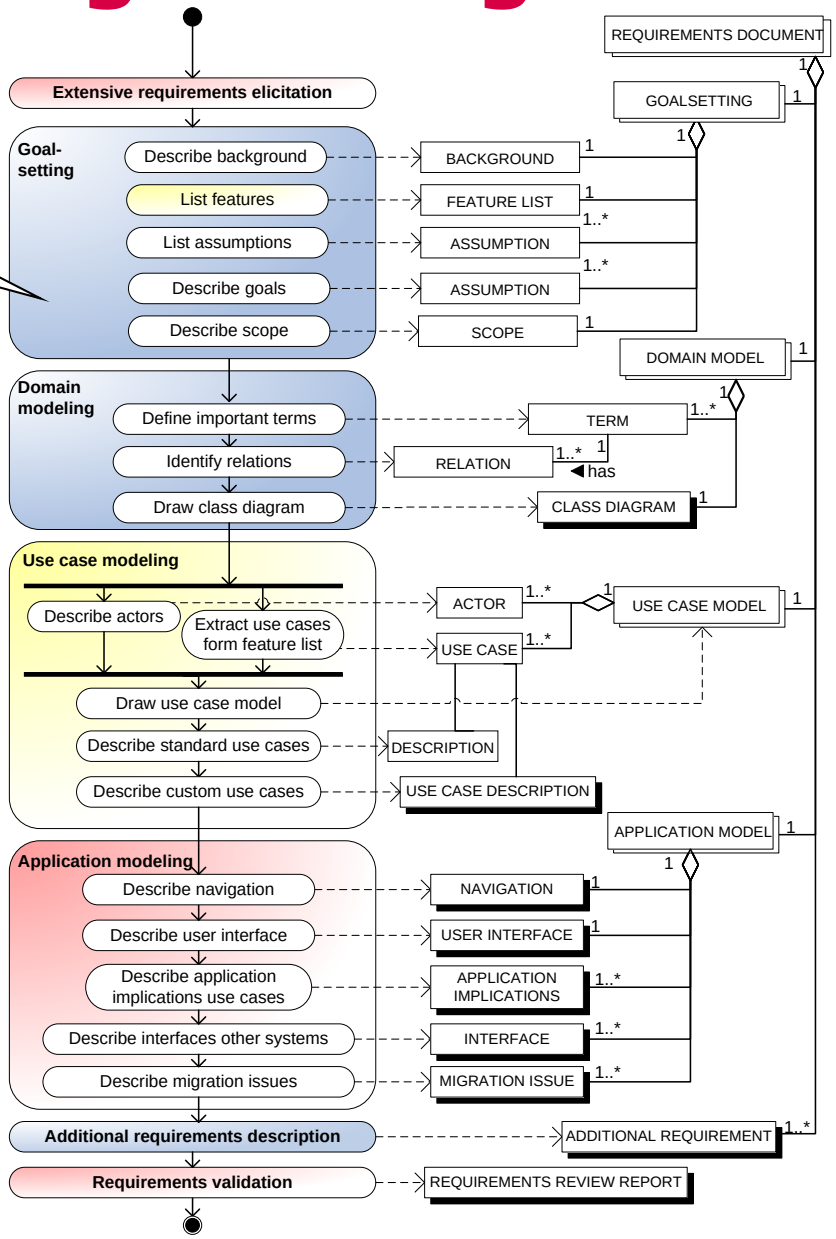
PDD example



Risk workflow in UML-Based Web Engineering

GX web engineering method

Use of color to show origin of the method fragments

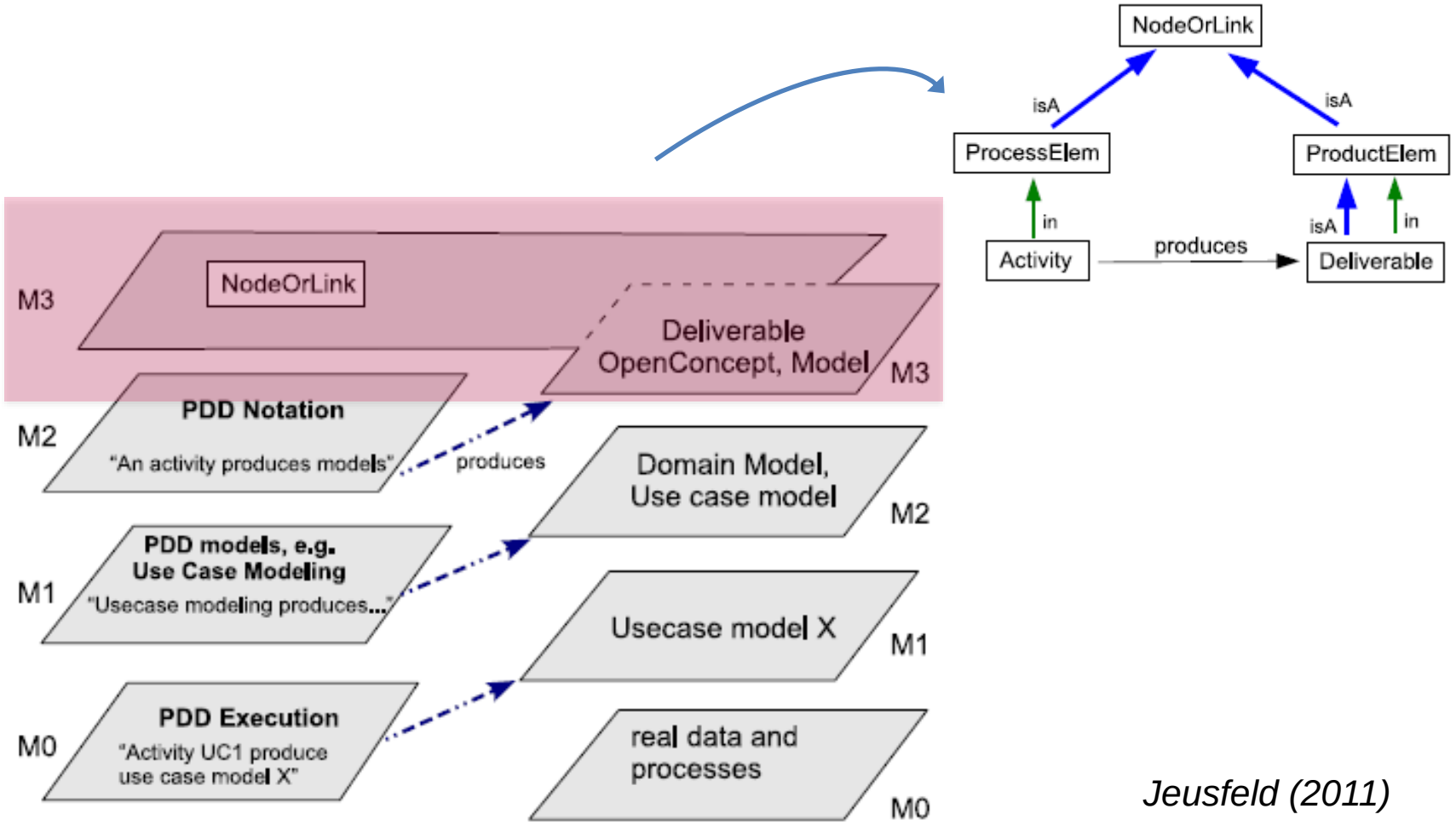


GX method

Unified Process

UML-based Web Engineering

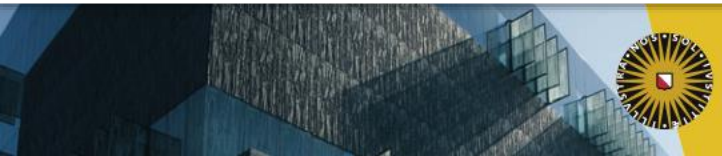
Meta-modeling perspective of PDDs



Jeusfeld (2011)

Agenda

- Data modeling assignment
- Introduction to meta-process modeling
- PDD notation
- Meta-modeling assignment



Assignment C

Objective:

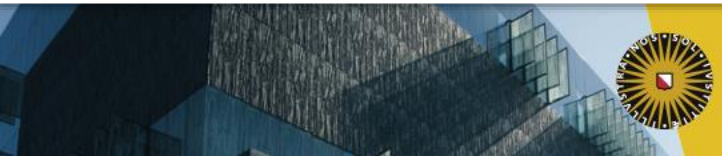
Create a documented Process-Deliverable Diagram (PDD)

Contents of the assignment:

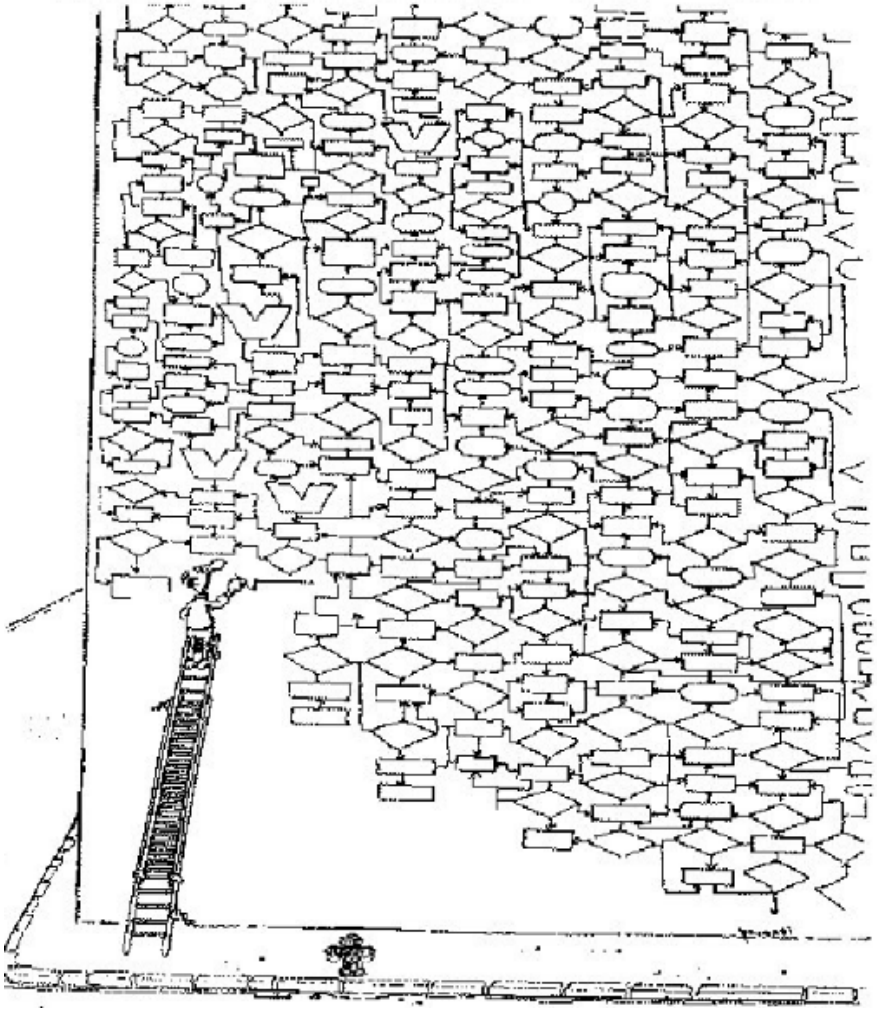
1. Topic name + some introductory sentences
2. Process-deliverable diagram
3. Explanations of the diagram
4. Activity table with the explanations of the activities
5. Concept table with definitions & source
6. List with references that you used for your PDD and table of concepts

Instructions meta-modeling

- Follow the instructions of the slides (lectures 1, 2, 3 & 4) and the paper “Meta-modeling for situational analysis and design methods”
- We use Visio for creating the PDDs. A Visio PDD stencil can be downloaded from the website. Alternatively, tools like draw.io or Lucidchart can be used.
- Take a look at the examples from last year

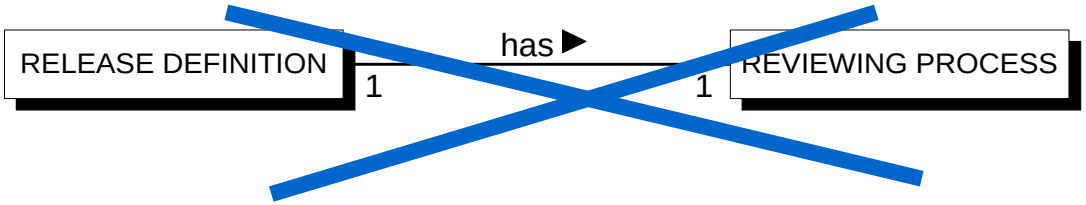


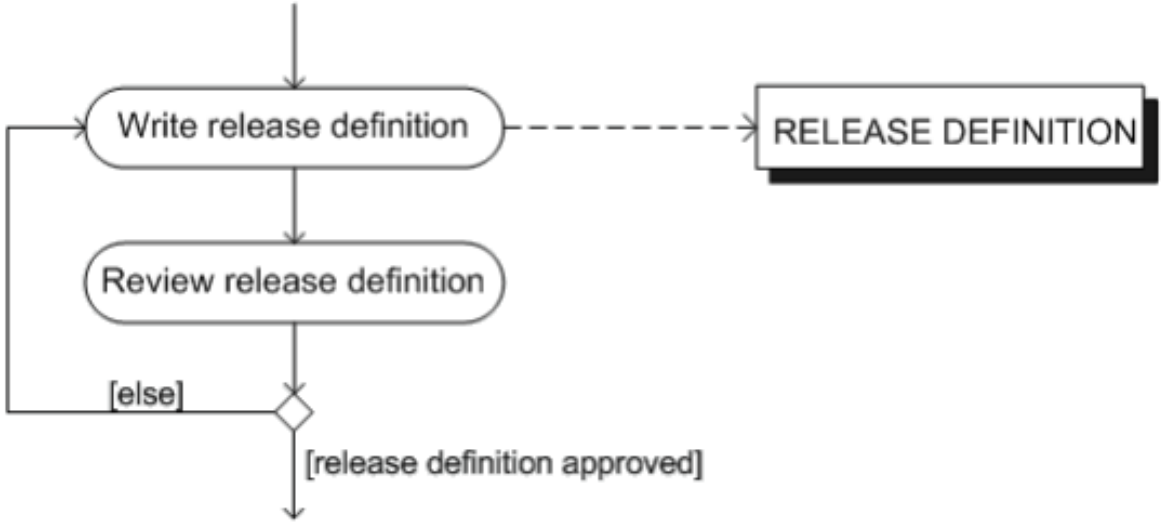
Find balance…



Frequent errors in meta-modeling (1)

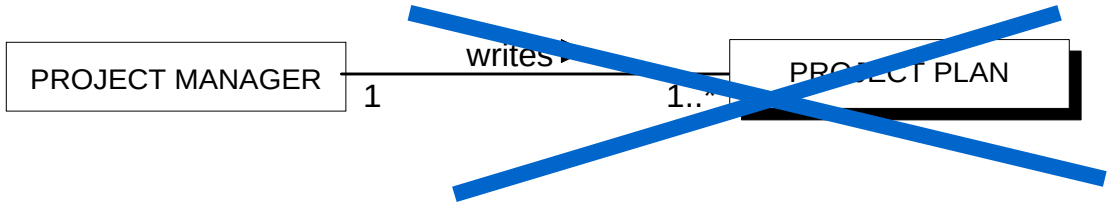
- Processes modeled in the deliverable side, e.g.:

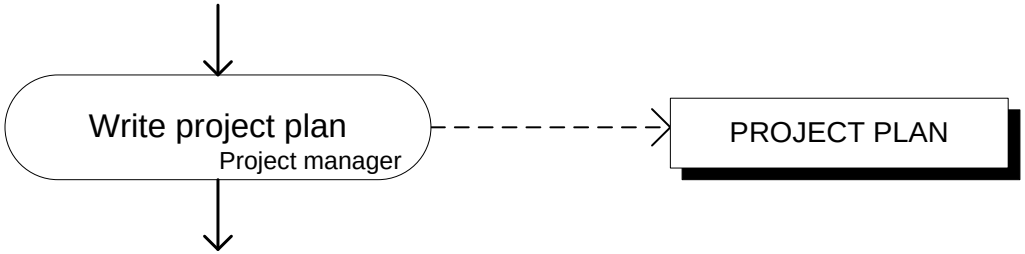




Frequent errors in meta-modeling (2)

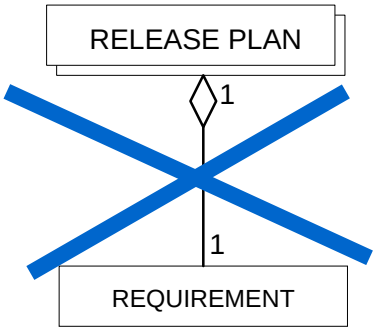
- People modeled in the deliverable side, e.g.:

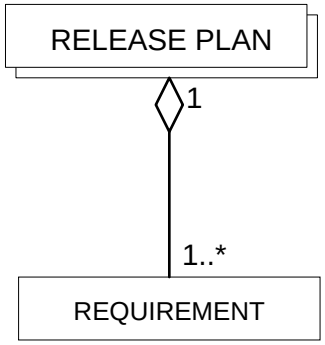


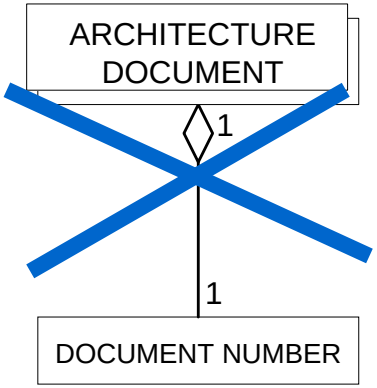


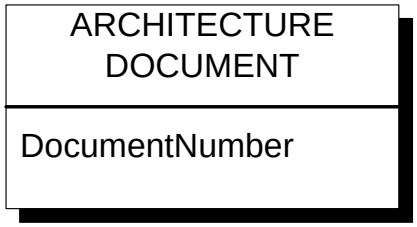
Frequent errors in meta-modeling (3)

- One-on-one relations that should have been modeled differently:









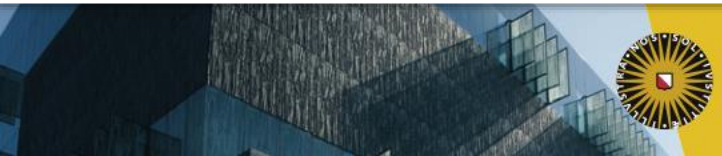
And many more…

- Mixing generalization with aggregation or association
- Forgetting roles, multiplicities, and naming of associations
- Use of unreliable sources in the definitions in the deliverable table → e.g.: www.deltatitlecorp.com
- Activity names too long
  - 'Use definition study to create functional design' should be 'Create functional design'
- Activity names too vague
  - 'Check' should (for example) be 'Review requirements document'

Deadline

- Deadline: March 15, 18h00.
  - Carefully read the description of Assignment C on the website
  - Submission in Revisely

- Good luck!

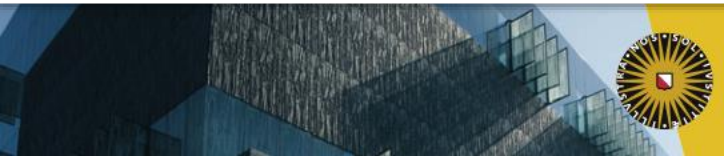


Mid-term exam

- The first three e-lectures and the related discussion sessions and slides (including the Introduction lecture in week 6 but excluding the Assignment Explanation lecture)
- Weerd, I. van de, Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. In M.R. Syed and S.N. Syed (Eds.), *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications (pp. 38-58)*. Hershey: Idea Group Publishing.
- Closed book
- Digital multiple-choice exam

- Date: 4 March 2019
- Time: 13h30-16h00
  (extra time students have until 16h30)
- Location: Educ-Alfa

QUESTIONS?



References

Heineman, G., Botsford, J.E., Caldiera, G., Kaiser, G.E., Kellner, M.I., & Madhavji, N.H. (1994). Emerging Technologies that Support a Software Process Life Cycle. IBM Systems Journal 32(3), 501-529.

Kurtev, I. (2007). Metamodels: Definitions of structures or ontological commitments? Proceedings of the Workshop on Towers of Models. Zurich, Switzerland, 53-63.

van de Weerd, I., & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. In M.R. Syed and S.N. Syed (Eds.), *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications* (pp. 38-58). Hershey: Idea Group Publishing.