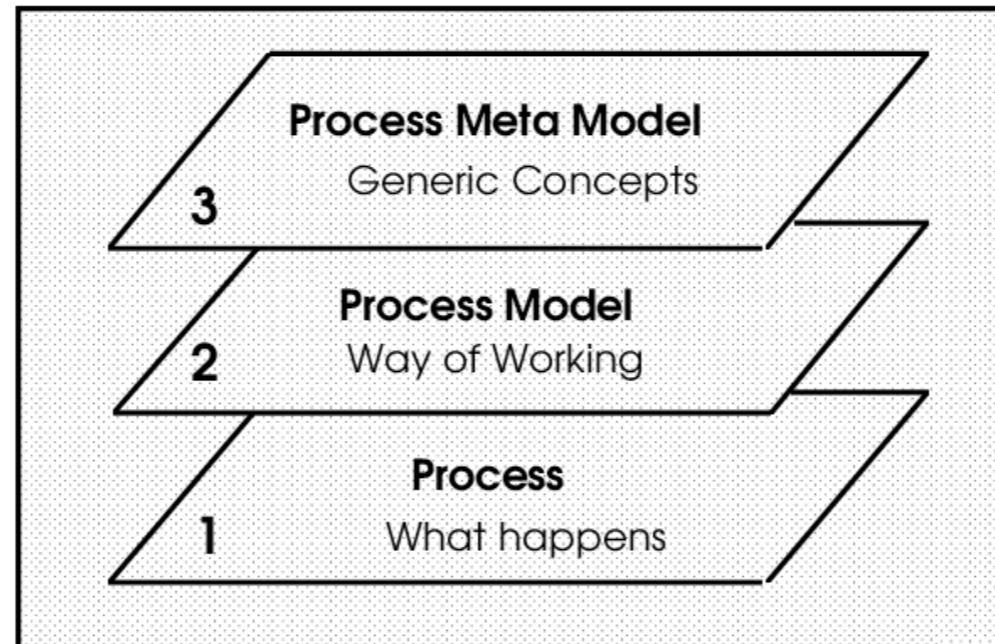


Objective



To lay the foundation of business process modelling through abstraction concepts

Ch. 3 of Business Process Management: Concepts, Languages, Architectures

Business process

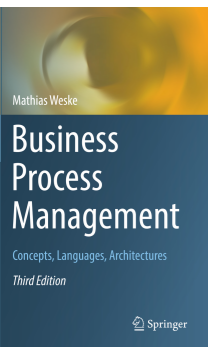
Definition: a **business process** consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal.

Orchestration

Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations.

Collaboration / Choreography

- *Weske*



Business process management

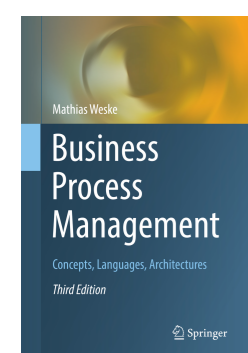
Definition: **business process management**

includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes.

- *Weske*

We need **explicit representation** of business processes, their **tasks** and the **execution constraints** between them

Business processes can then be subject to **analysis, improvement, and enactment**



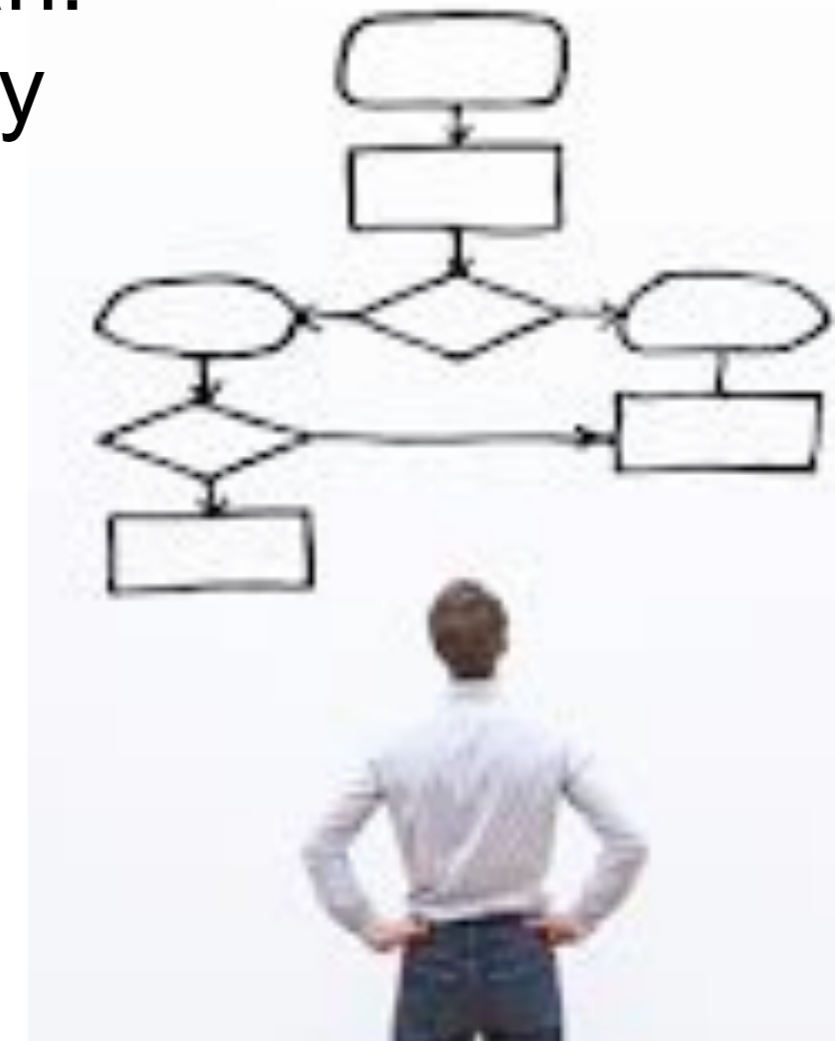
Modelling

From informal textual descriptions (requirements) to a particular business process modelling notation

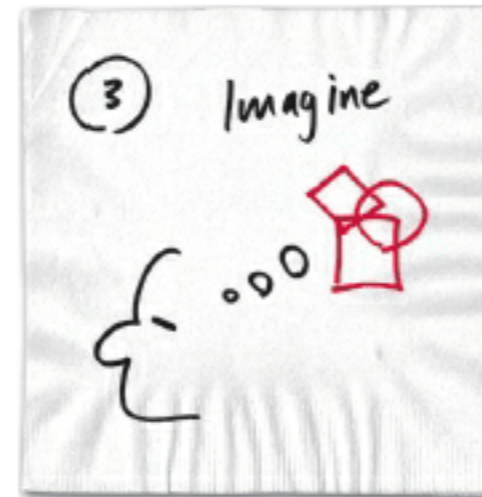
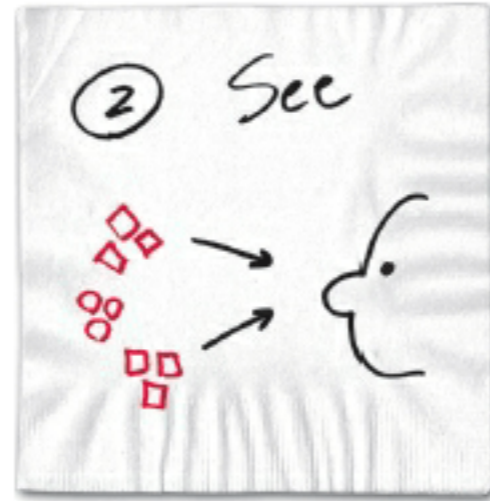
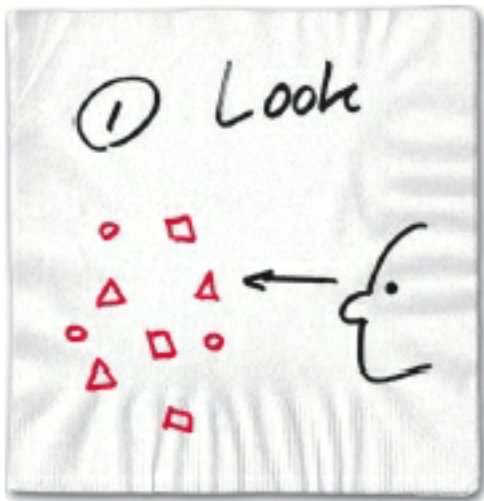
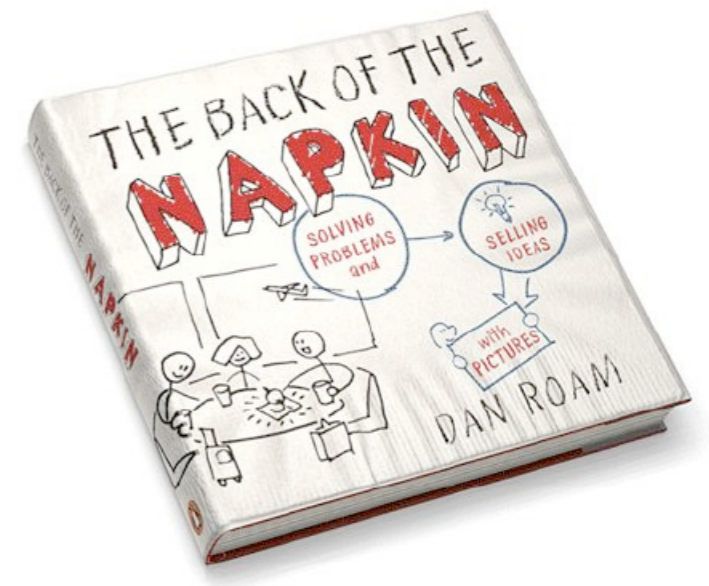
Explicit business process models expressed in a graphical notation facilitate communication, so that different stakeholders can:

communicate efficiently
refine processes
improve processes

No	Requirement	Use Case
1	To locate user's current location	Current Location
2	To display user's current location information	Display Current Location
3	To display error when couldn't locate location	Error Displaying Locations
4	To provide a search function for user to search for direction to another location	Search Direction
5	To display user's desired location direction	Display Direction
6	To provide user the floor plans of the building	Floor Layout
7	To provide the list of floor layout for user to select	Select Floor
8	To display the selected floor layout by the user	Display Floor Layout
9	To provide user all the stores information in the building	Store List
10	To provide the list of store categories for user to select	Select Category
11	To display the list of stores under the selected	Select Store
12	To display the selected store information	Display Store Information
13	To display the first interface of	Home



Look, see, imagine, show



Representing processes

Visual representations:

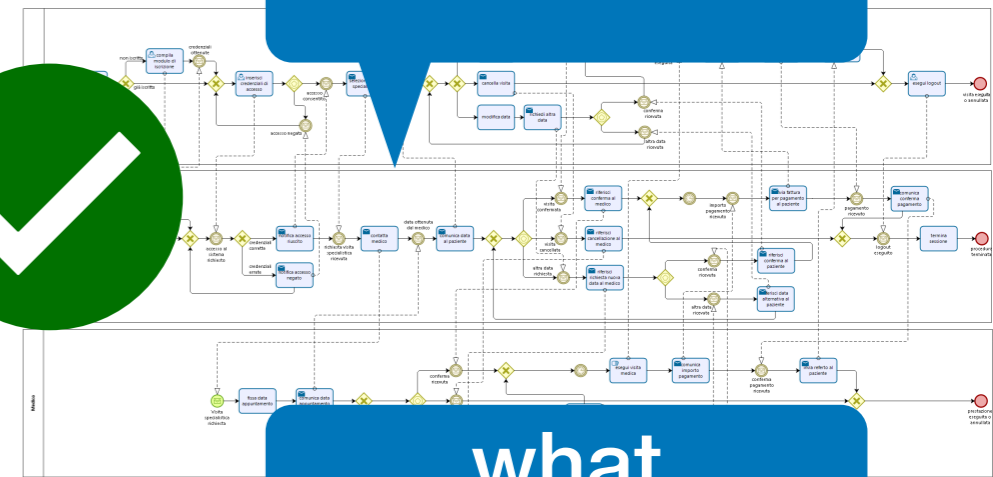
diagrams and charts

understandable by humans

(informal, intuitive, BPMN, EPC, BPEL)



what we see



what machines see

Languages:

unambiguous machine syntax

(process dialects, XML schemes)

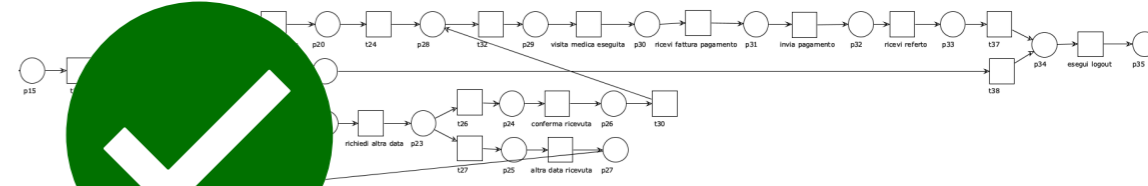


```
1 <?xml version="1.0"?>
2   xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/
   instance" id="_2015_1207470" targetNamespace="http://www.bizagi.com/
   9011207470" xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL">
   Id_4184d428-51f4-4a5f-a66c-4b24ae1da1b3" name="Processo principale">
   n />
   Id_b106ba3a-1bbb-4efd-bece-dc9cda12eeeb" />
   Id_15079acc-09ed-4adf-b44f-1578d60b82a8">
   tion />
   id="Id_ebce1c69-4d72-45f6-8b3e-2367442967ab" />
10 <intermediateCatchEvent id="Id_254c9cde-6b8c-42be-a151-55ed4e87b322" name="conferma
   ricevuta">
```

Models:

rigorous semantics for scientists

(automata, Petri nets, workflow nets)



what we analyse

Do you know XML?

eXtensible Markup Language:
file format for storing and transmitting data

XML tags represent the data structure and contain metadata

```
<?xml version="1.0" encoding="UTF-8"?>  
<note>  
  <from>Alice</from>  
  <to>Bob</to>  
  <heading>Reminder</heading>  
  <body>Don't forget to buy oranges!</body>  
</note>
```


BPMN vs .bpmn



```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn:process id="Process_0pjif87">
  <bpmn:startEvent id="StartEvent_1">
    <bpmn:outgoing>Flow_0u05dpy</bpmn:outgoing>
  </bpmn:startEvent>
  <bpmn:task id="Activity_11zhm01" name="Run">
    <bpmn:incoming>Flow_0u05dpy</bpmn:incoming>
    <bpmn:outgoing>Flow_17t5zjm</bpmn:outgoing>
  </bpmn:task>
  <bpmn:sequenceFlow id="Flow_0u05dpy" sourceRef="StartEvent_1" targetRef="Activity_11zhm01" />
  <bpmn:endEvent id="Event_1t0u7im">
    <bpmn:incoming>Flow_17t5zjm</bpmn:incoming>
  </bpmn:endEvent>
  <bpmn:sequenceFlow id="Flow_17t5zjm" sourceRef="Activity_11zhm01" targetRef="Event_1t0u7im" />
</bpmn:process>
</note>
```

Models

A **model** is a simplified representation of reality

*"Essentially all models are wrong,
but some are useful"*
(George P. Box)

How to cope with complexity?
(Which capacity is essential to build models?)

Abstraction!

Abstraction

To derive general rules and concepts
from specific examples of some phenomenon,
by selecting only the aspects which are relevant
for a particular purpose

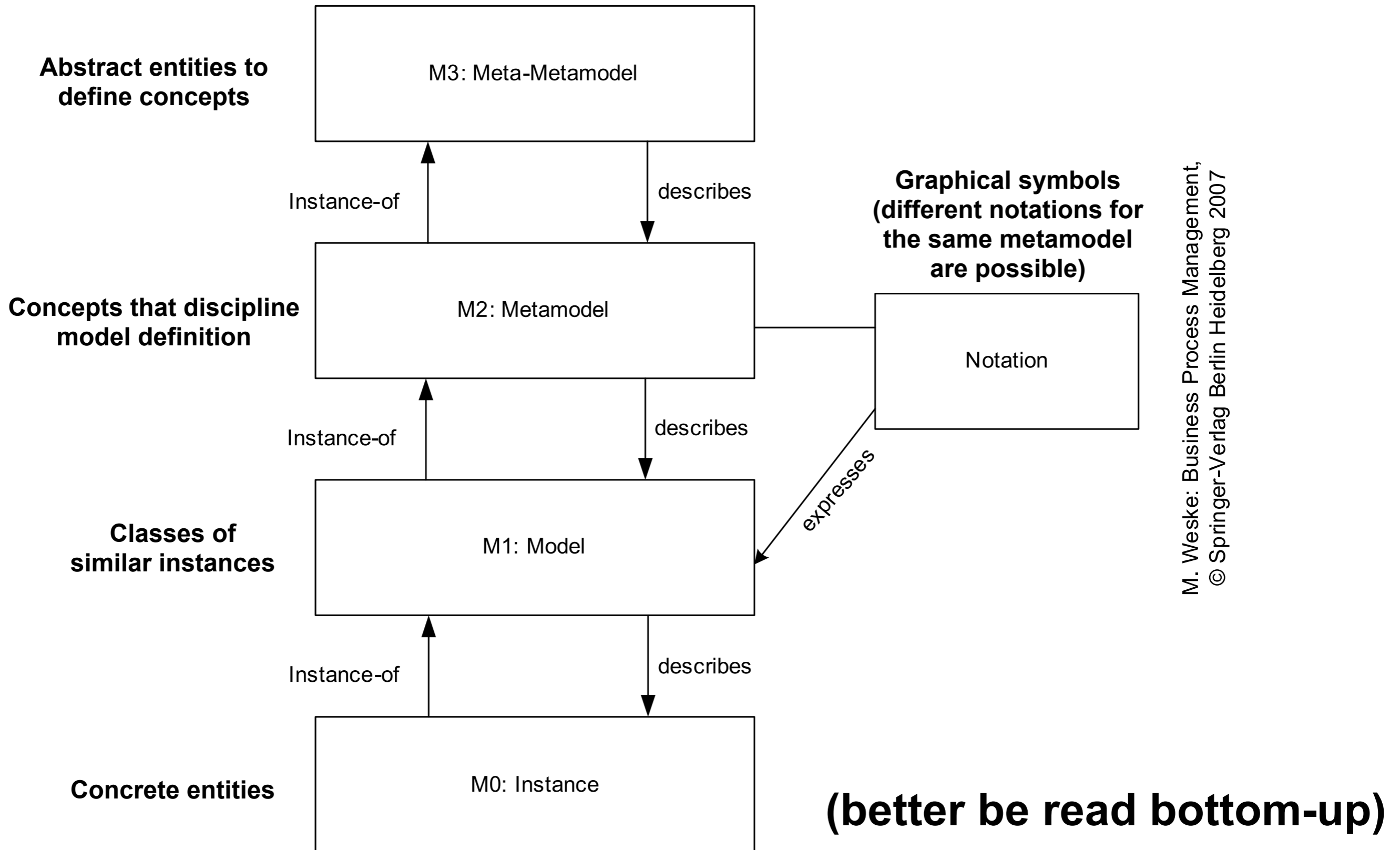
Horizontal: separation at different modelling levels

Aggregation: separation at different granularity levels

Vertical: separation at different subdomains

Horizontal Abstraction (modelling levels)

Horizontal abstraction




From your forms






A general-purpose *visual modeling language* that is intended to provide a standard way to visualize the design of a system

5. Do you know the graphical notation for UML class diagrams or for Entity Relationship diagrams?

[Altri dettagli](#)

 Dati analitici

 Yes	21
 Just a little	13
 No	25

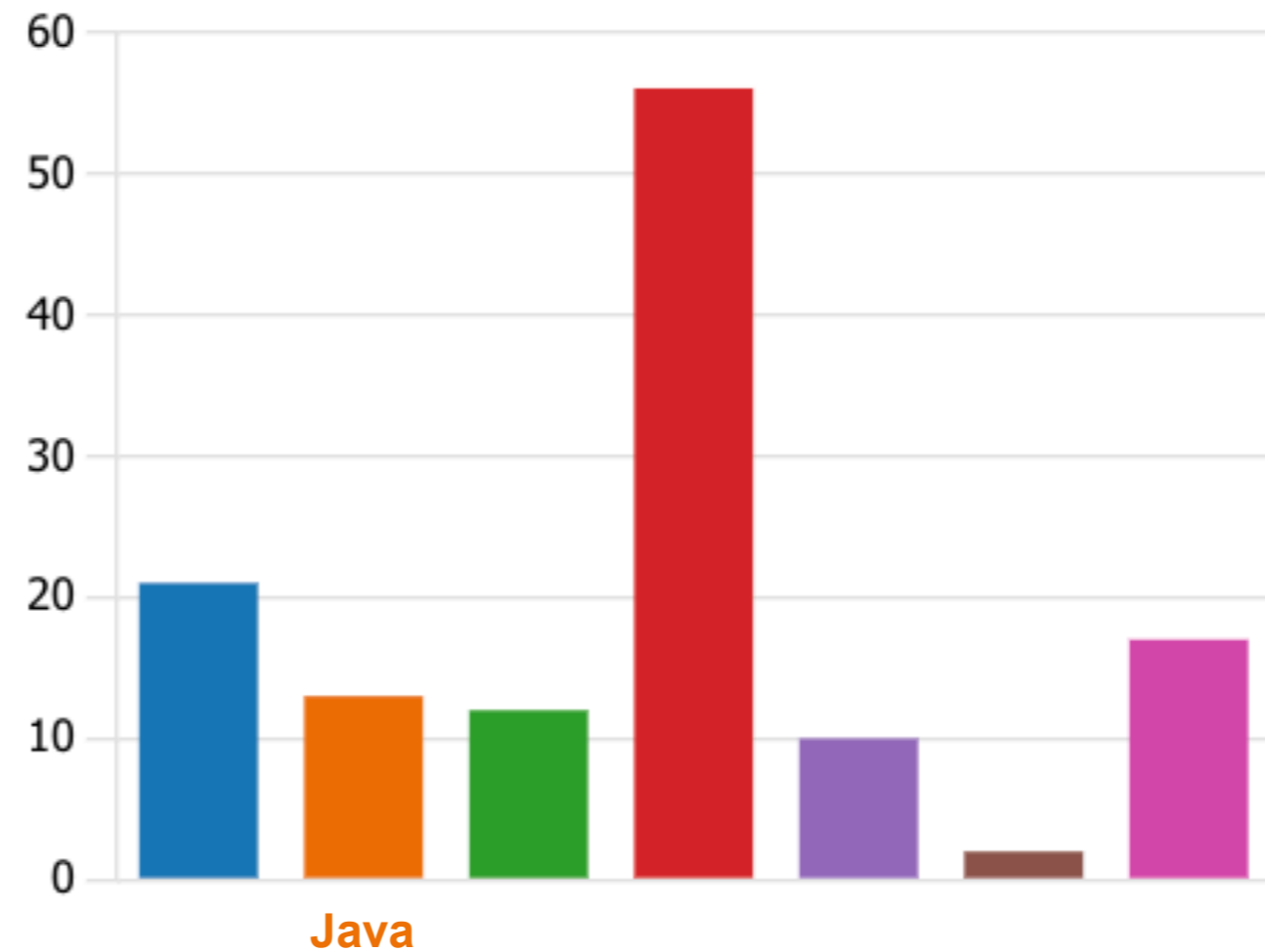


From your forms

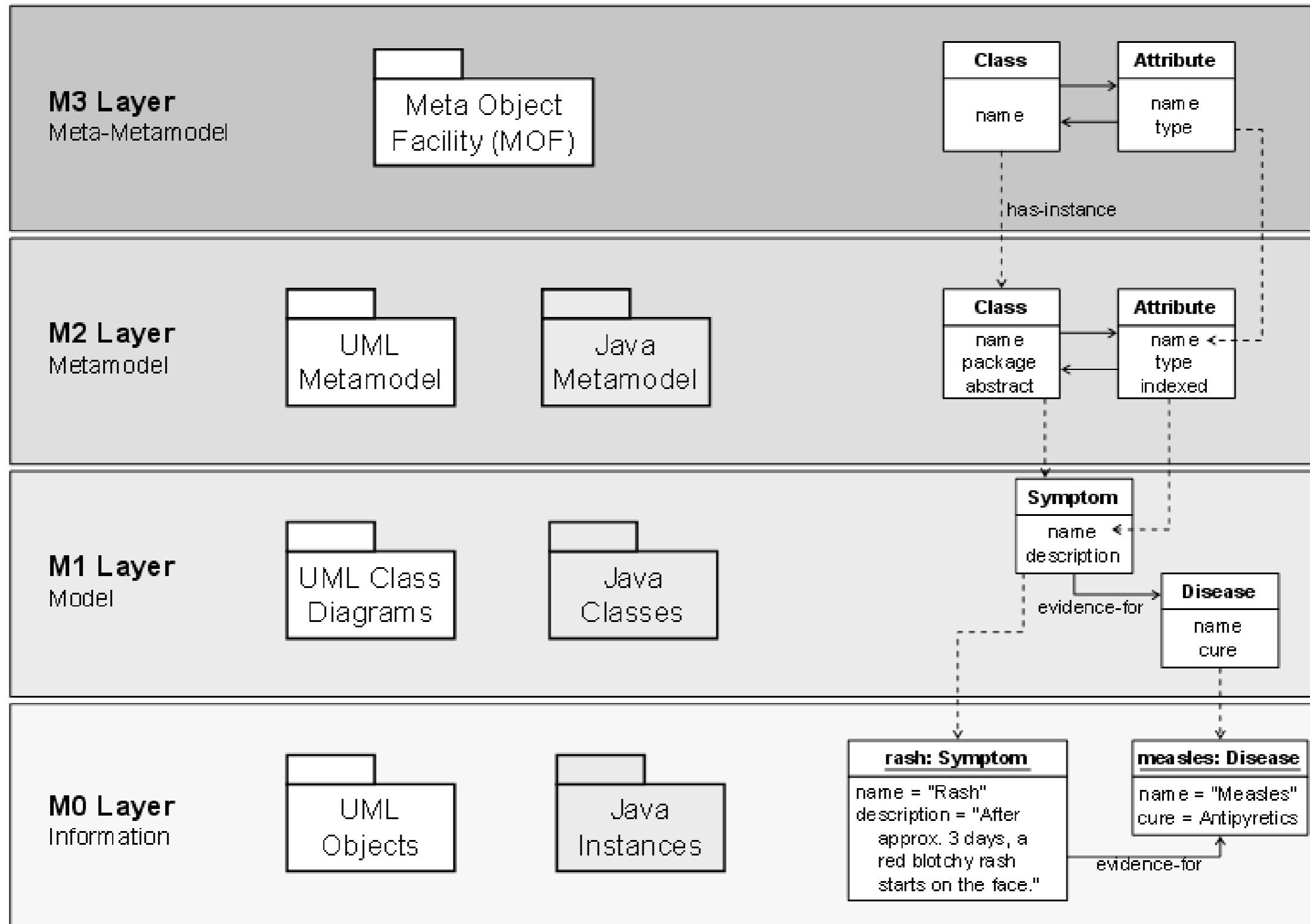
7. Check the programming languages you are familiar with:

[Altri dettagli](#)

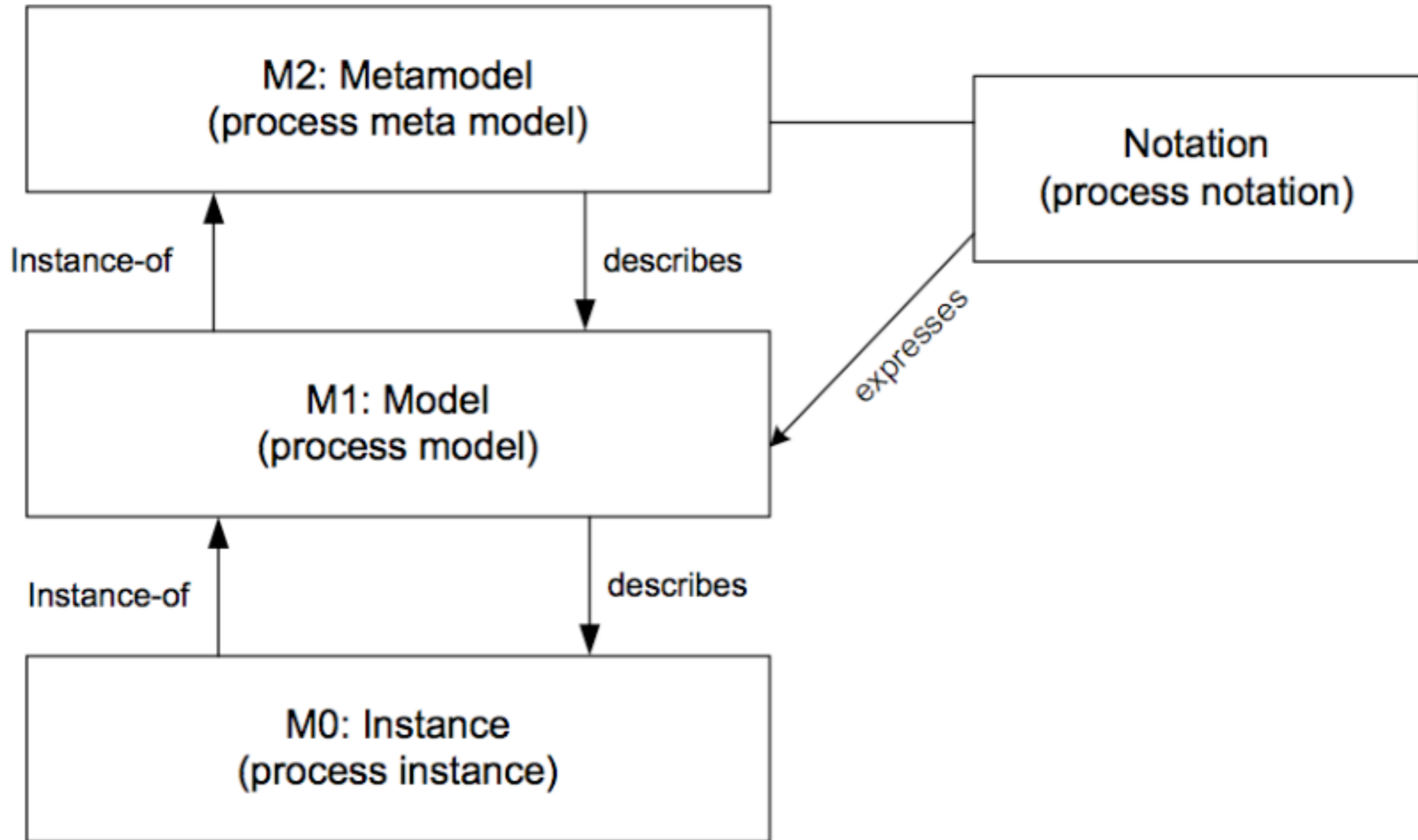
<input type="radio"/>	C, C++, C#	21
<input checked="" type="radio"/>	Java	13
<input type="radio"/>	Javascript	12
<input type="radio"/>	Python	56
<input type="radio"/>	PHP	10
<input type="radio"/>	ML, OCAML, Haskell	2
<input type="radio"/>	Altro	17



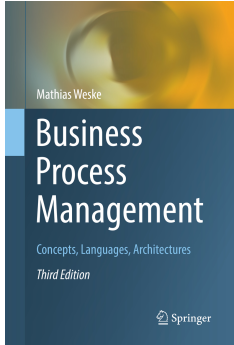
Ex: MOF metamodel (OMG)



Process models and process instances

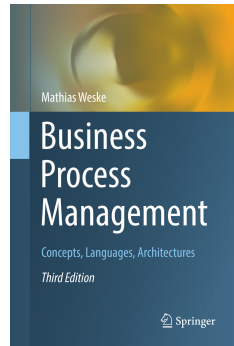


Business process models and instances



Definition: **business process model** consists of a set of activity models and execution constraints between them.

- *Weske*



Definition: **business process instance** represents a concrete case in the operational business of a company, consisting of activity instances.

- *Weske*

Models and instances



Enter
credit
request

Enter c.r.
(r017, Miller,
10000)

Enter c.r.
(r018, Brown,
15500)

Enter c.r.
(r019, McGraf,
12000)



Each activity model acts as a blueprint
for a set of activity instances

Each business process model acts as a blueprint
for a set of business process instances
(related to cases)

Abuse of notation

If no confusion is possible,
the term **activity** is used to refer
to activity models (tasks)
as well as activity instances

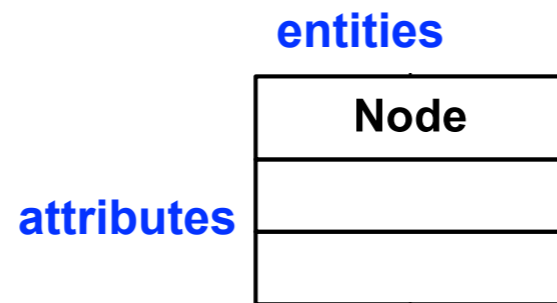
Analogously,
the term **process** is used to refer
to process models as well as process instances

Ingredients

start / end	nodes
tasks	nodes
join & split: concurrency	nodes
join & split: internal decisions	nodes
split: external decisions	nodes
links: causal and temporal dependencies	edges
responsibility: whole process / single tasks	?
information: data, parameters, ...	?
platform: bindings, services, ports, ...	?

A process metamodel (M2)

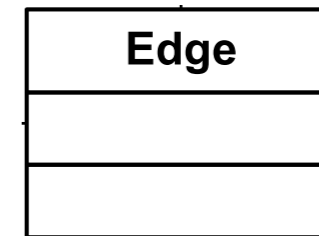
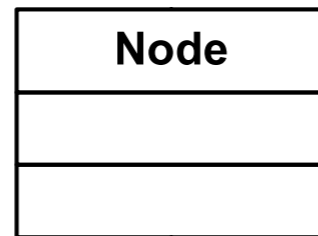
UML-like syntax



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

A process metamodel (M2)

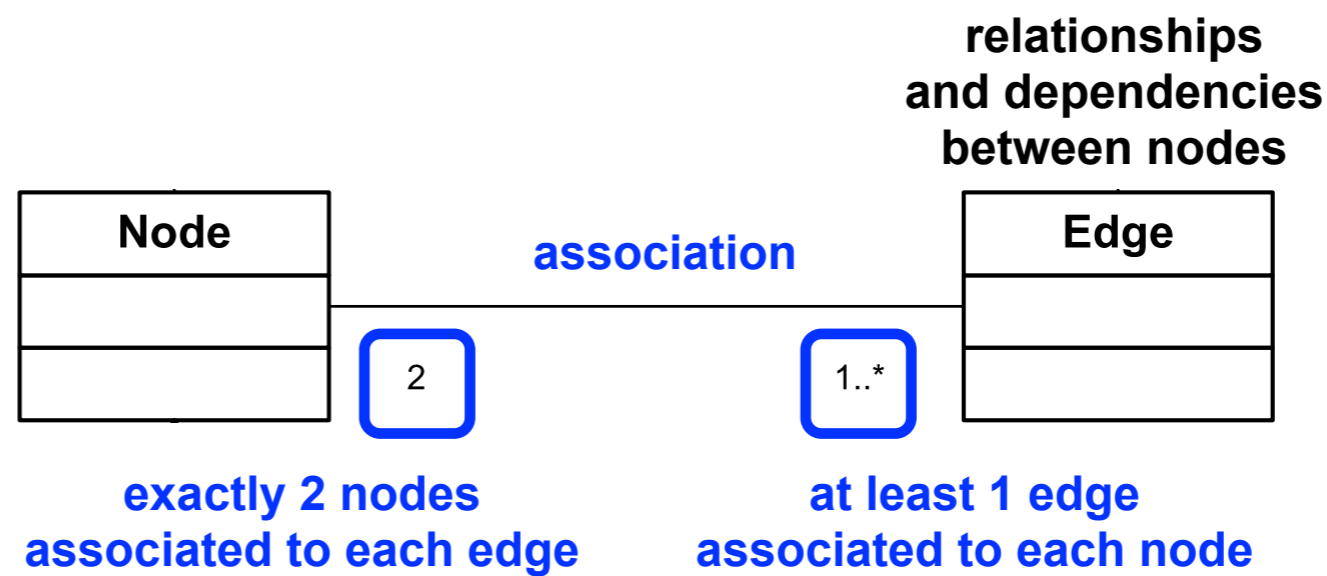
UML-like syntax



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

A process metamodel (M2)

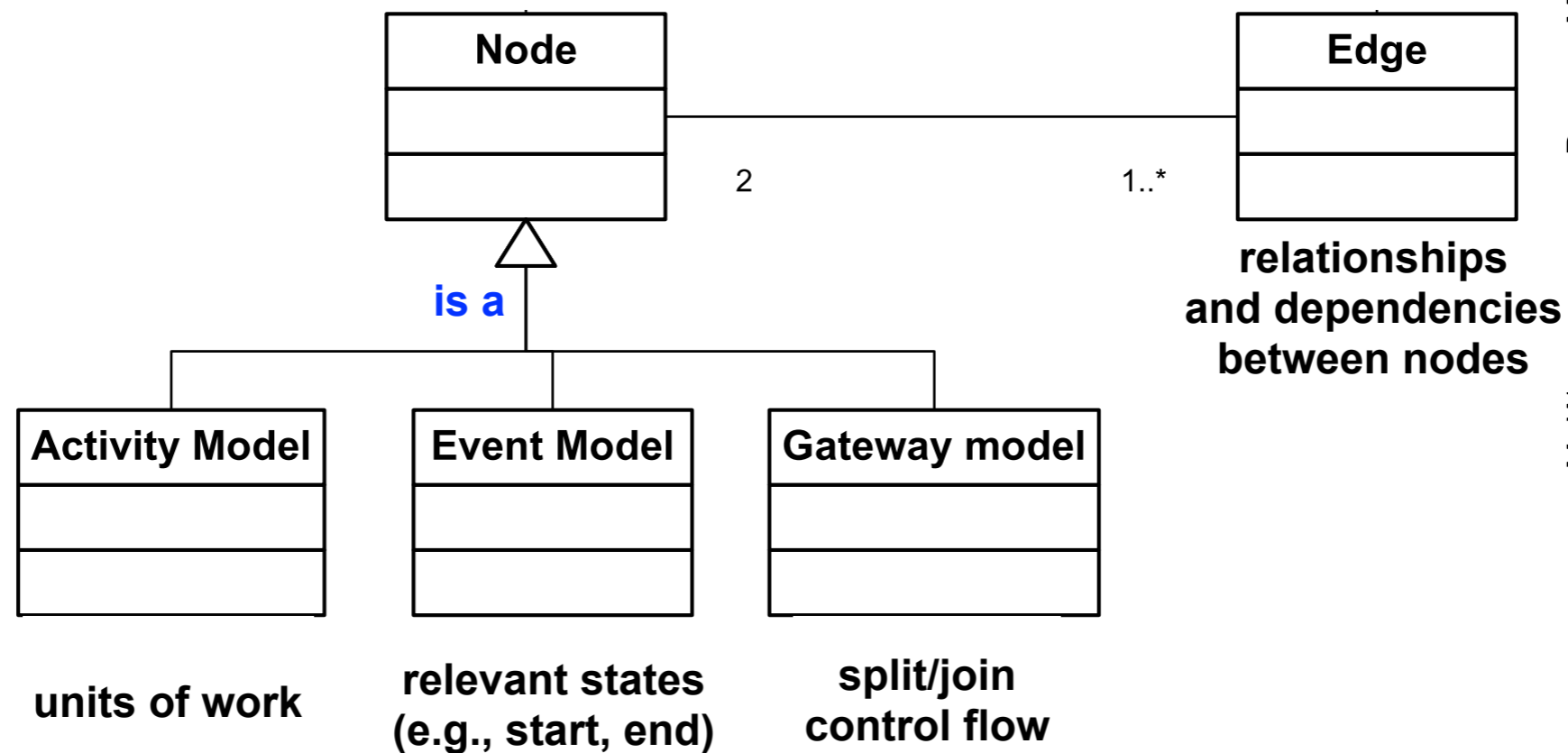
UML-like syntax



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

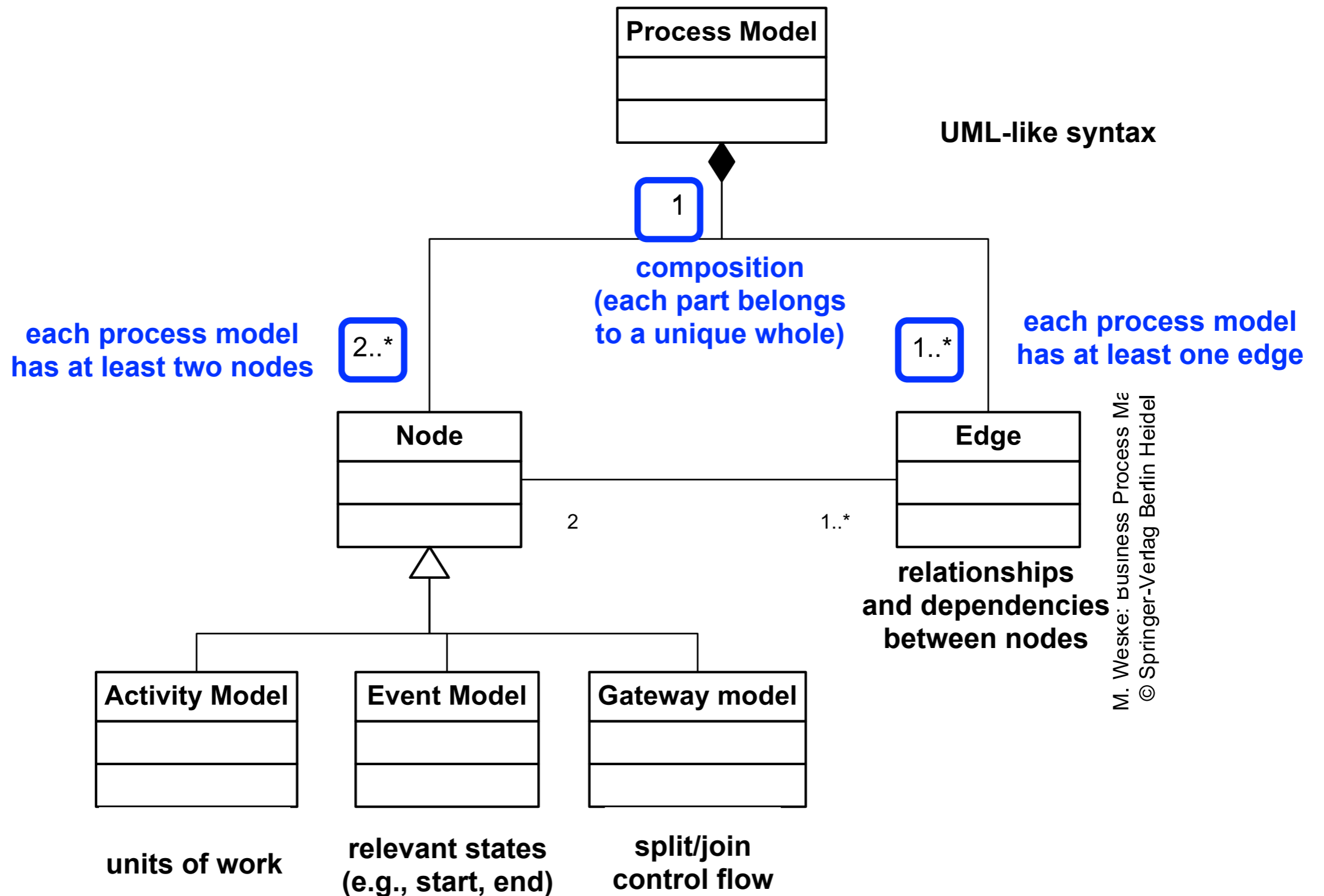
A process metamodel (M2)

UML-like syntax



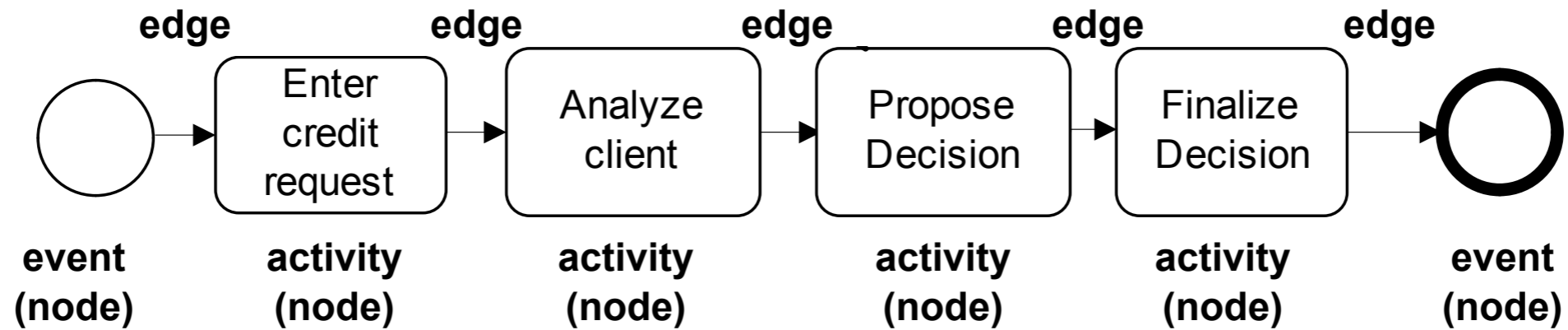
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

A process metamodel (M2)

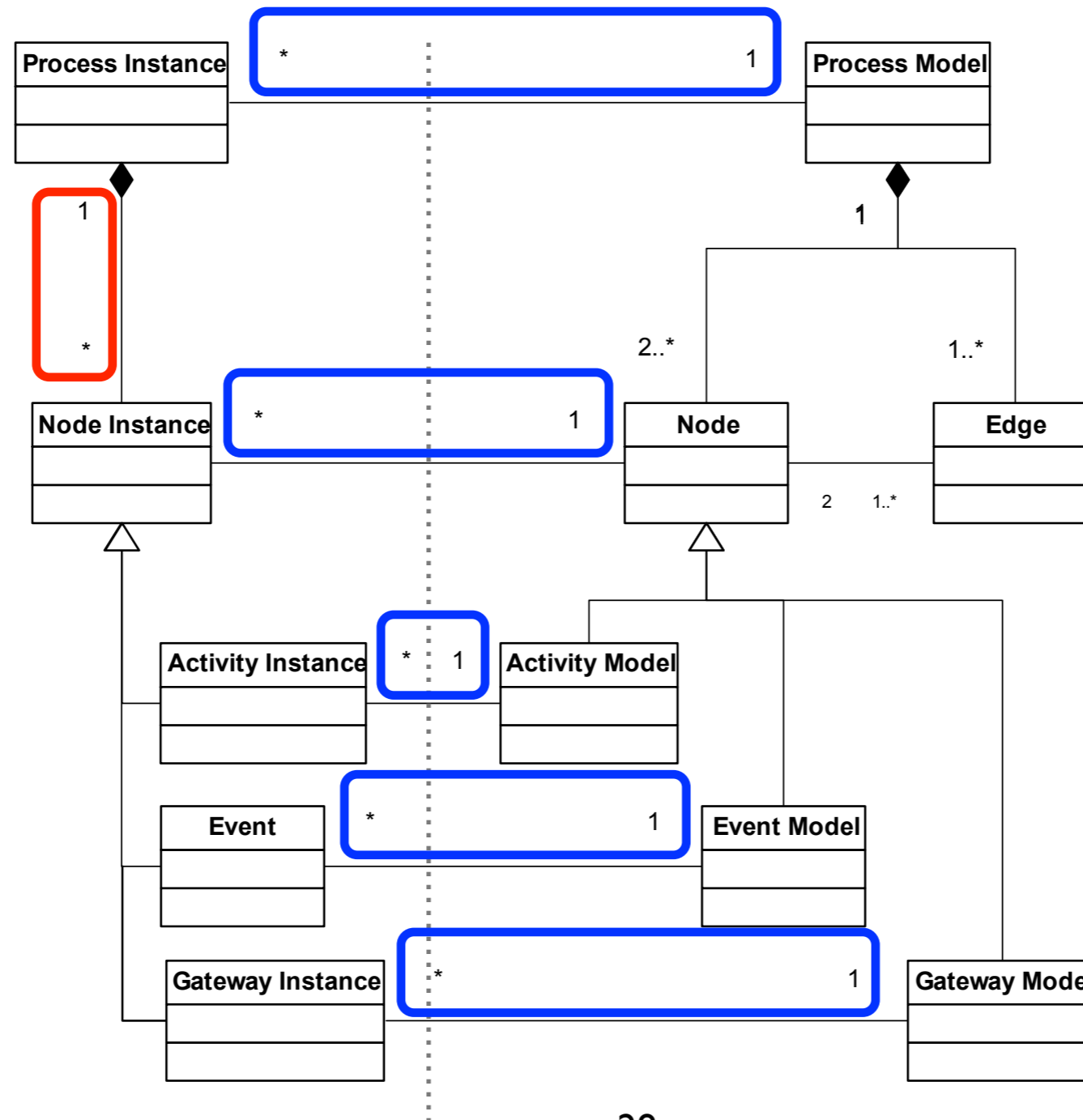


M. Weske: Business Process Management
© Springer-Verlag Berlin Heidelberg

A process model (M1)

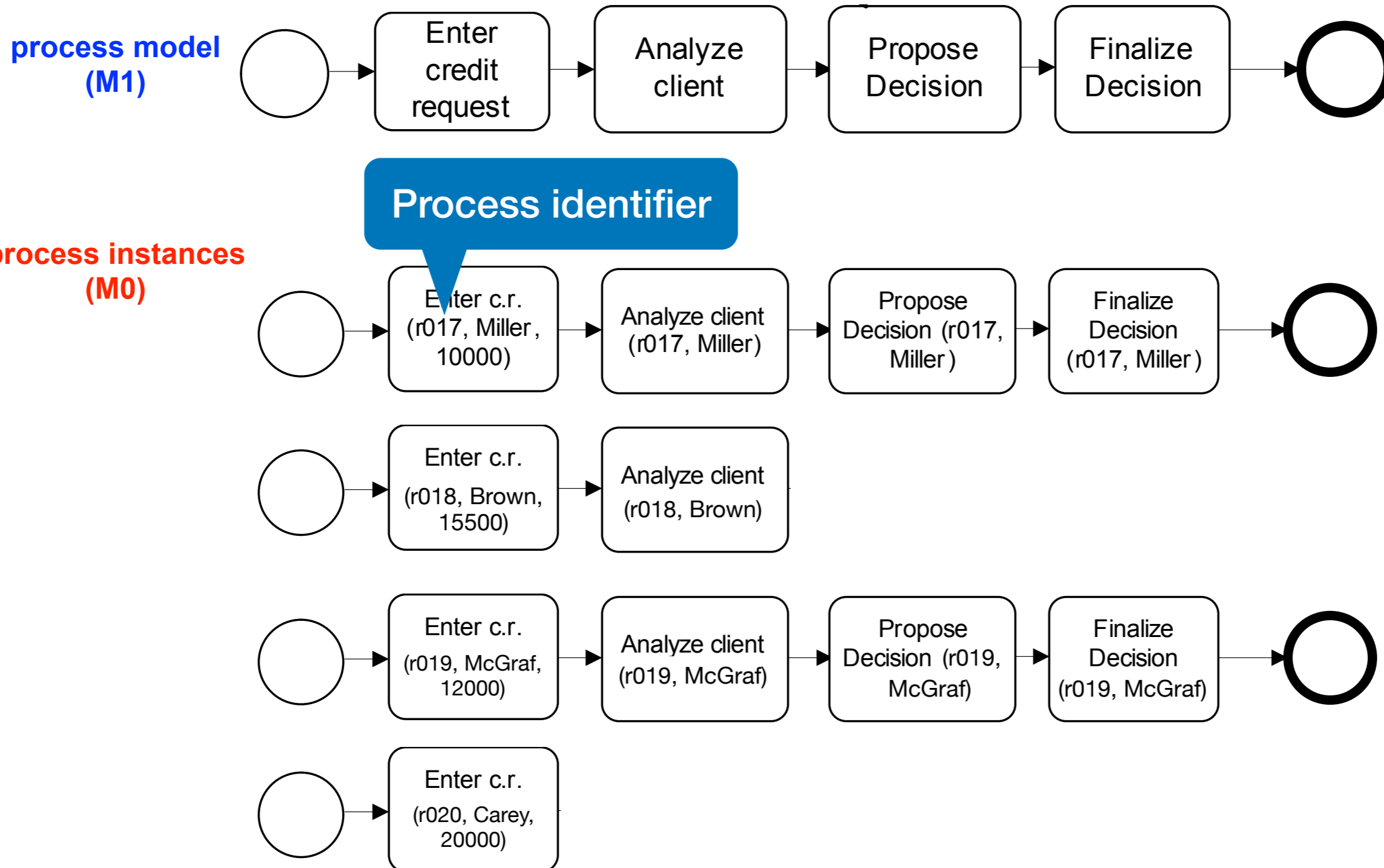


Process models and process instances (M2)

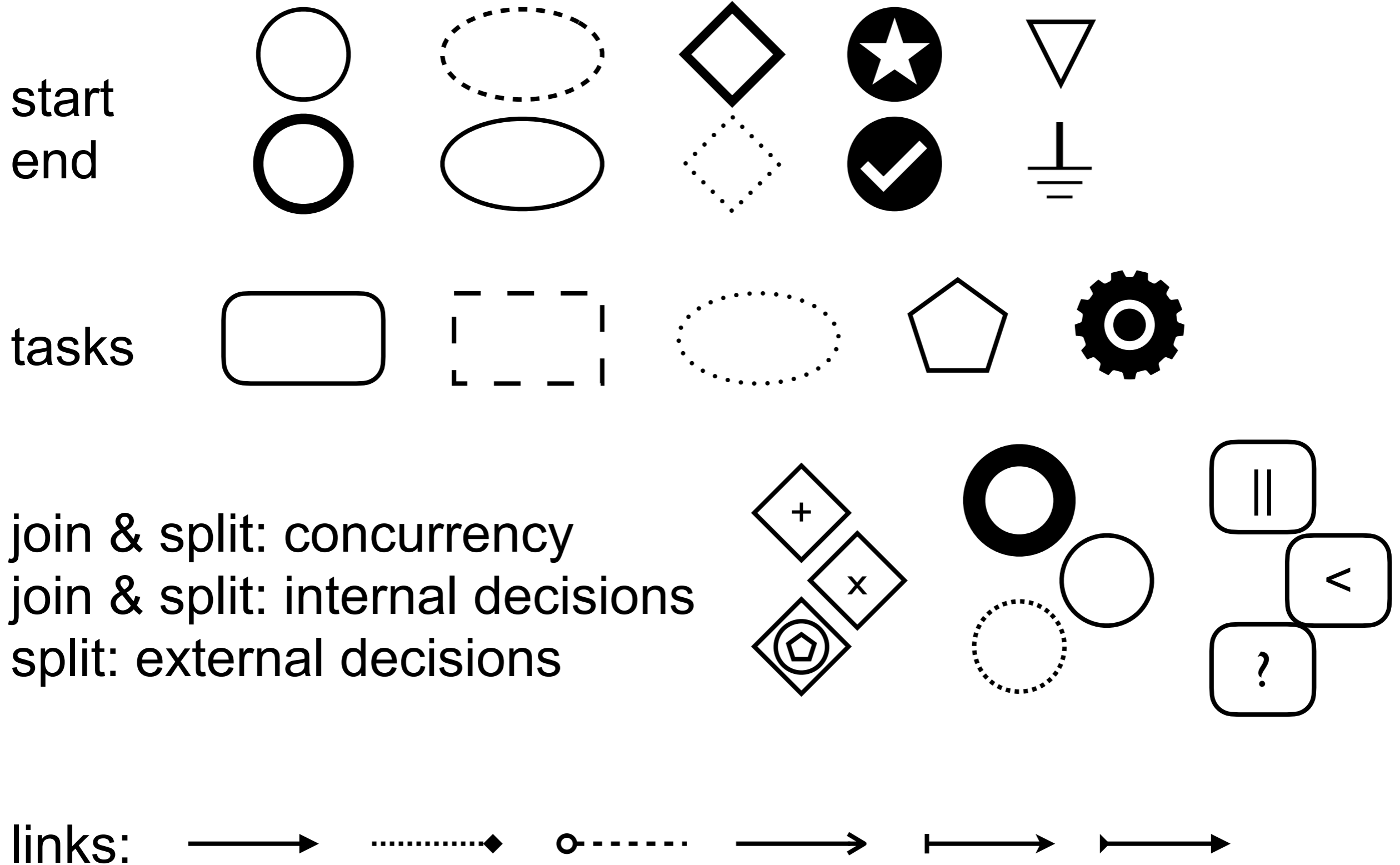


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

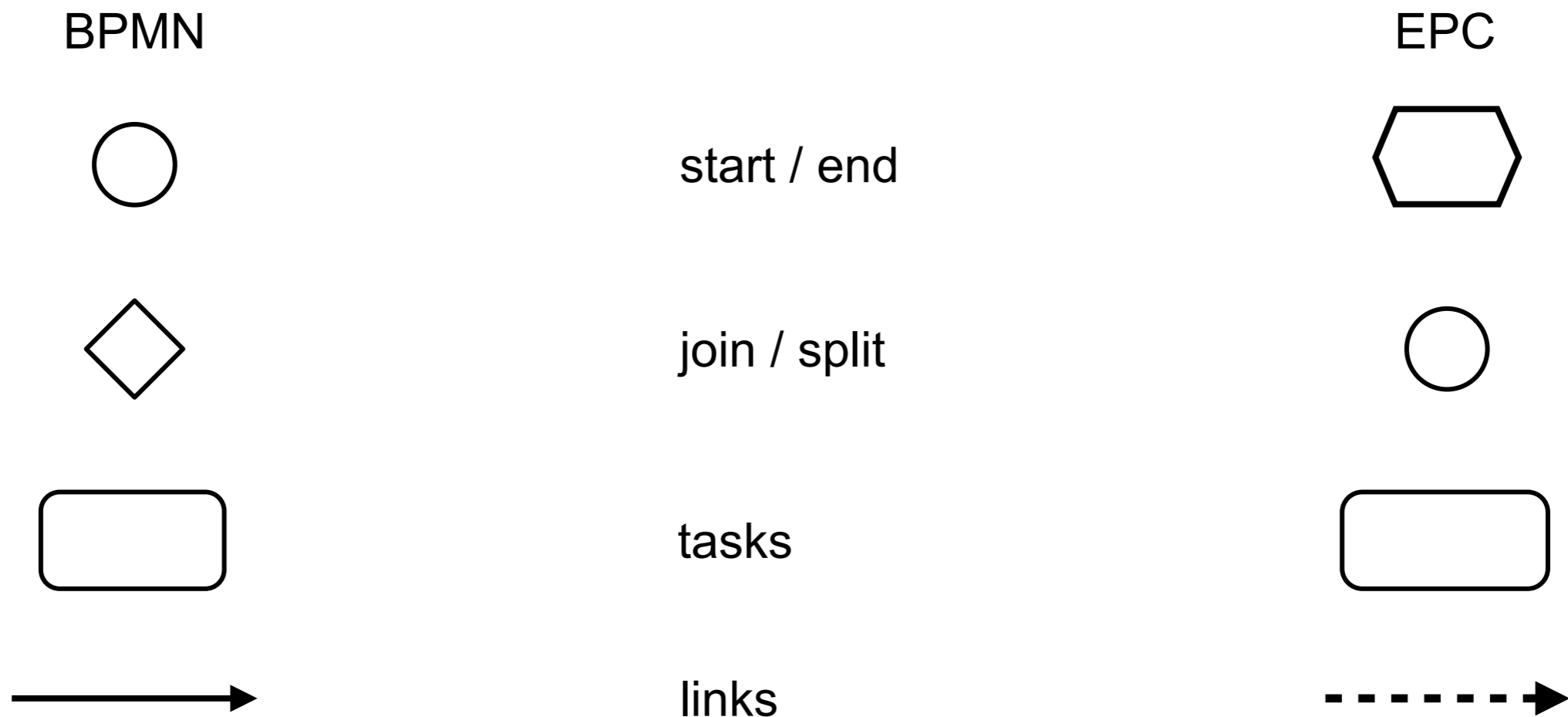
Some process instances (M0)



Let's invent our own notation



A sneak peek at BPMN and EPC



Aggregation Abstraction (granularity levels)

Aggregation abstraction

Multiple elements of a lower level of granularity can be grouped and represented by a single artefact at the higher level of granularity

Related to refinement / functional decomposition:

A single artefact at the higher level of granularity can be decomposed in multiple elements of a lower granularity level

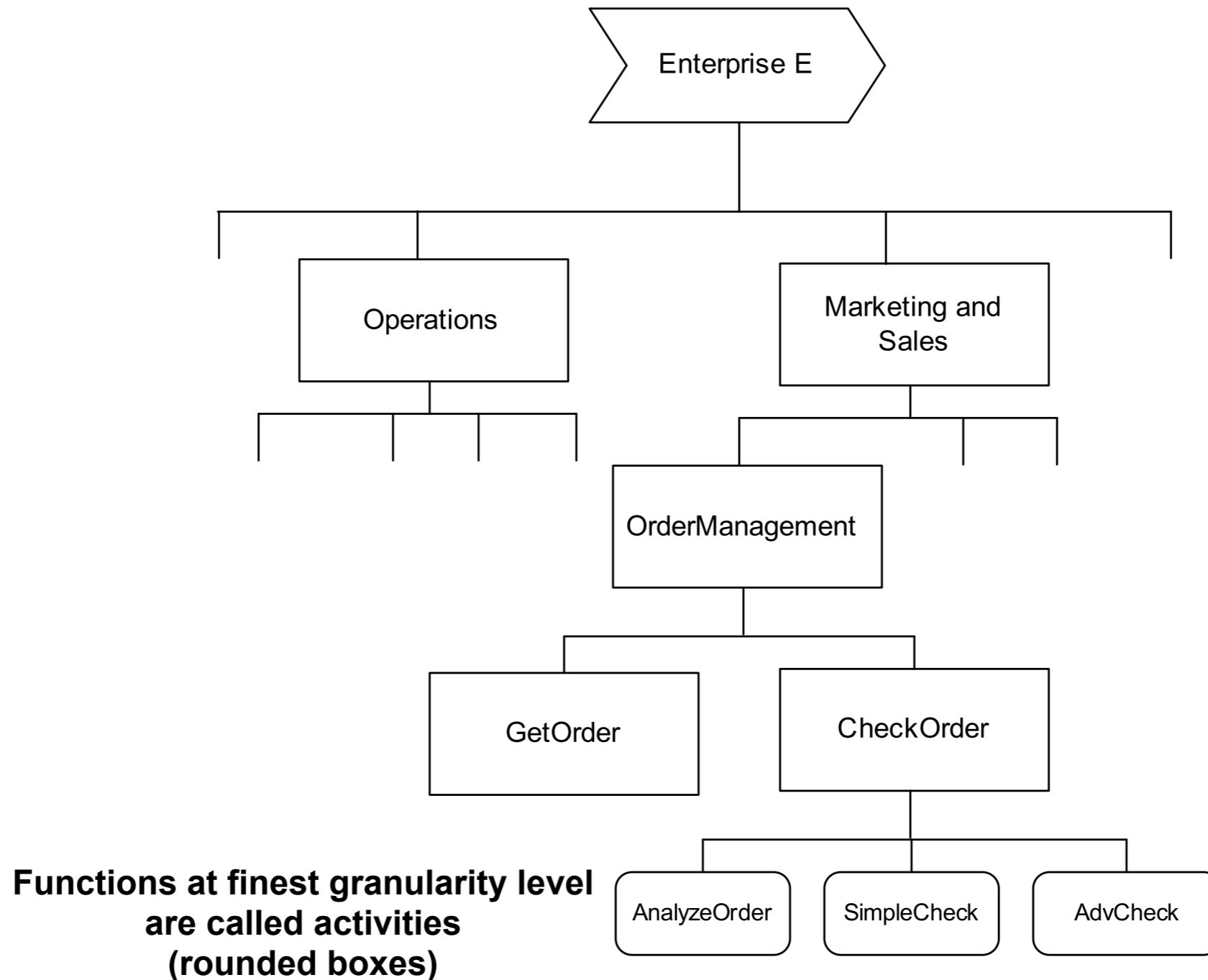
Different from horizontal abstraction, where all entities lie at the same level of granularity

High-level business functions

The value chain of a company has a rich internal structure, consisting of a set of coarse-grained business functions
(e.g. Order management, Human resources)

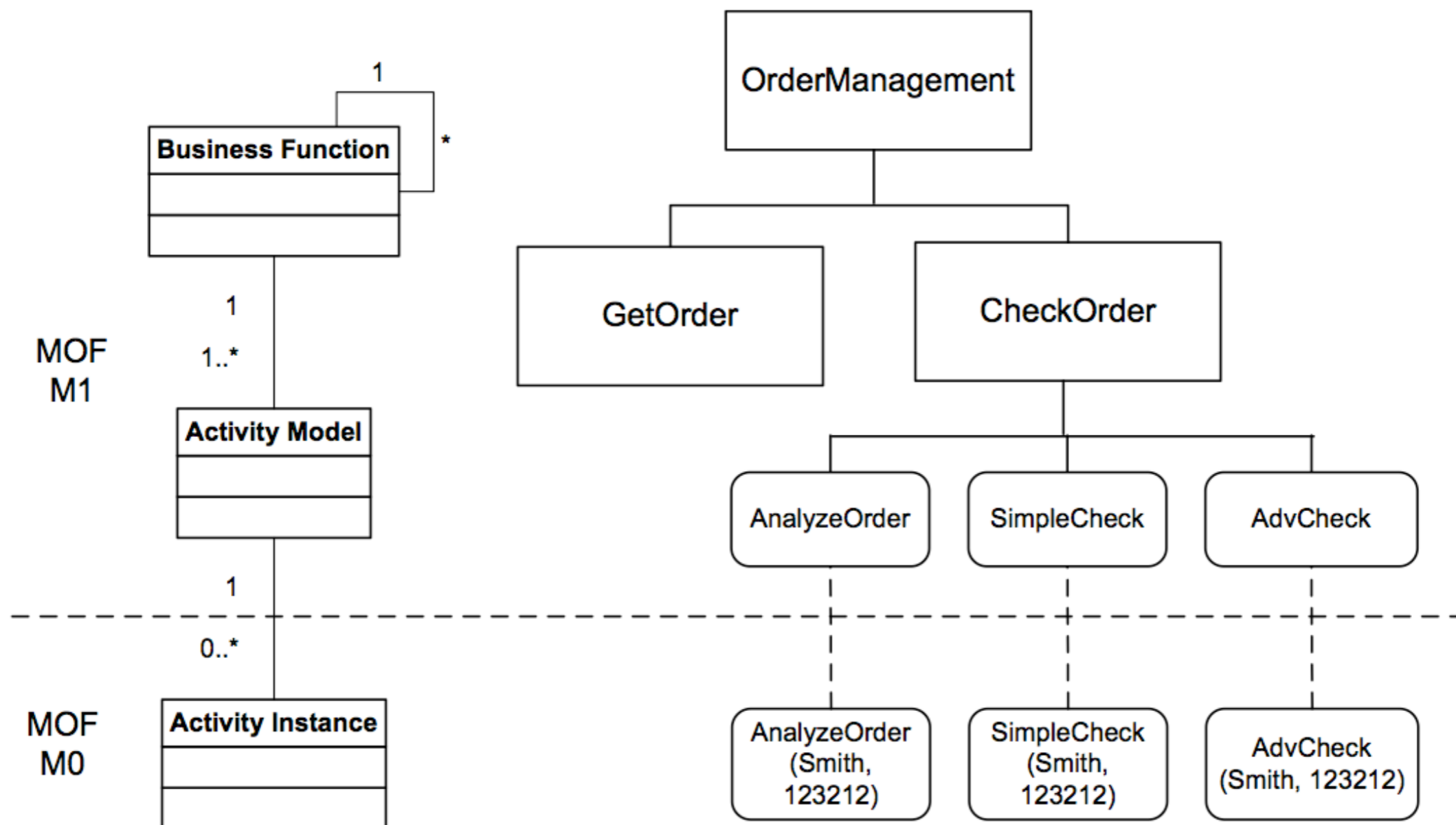
High-level business functions can be decomposed into finer-grained functions
(this is called **functional decomposition**)
(e.g. from ``Order management'' to ``storing'' and ``checking'' orders)

Functional decomposition

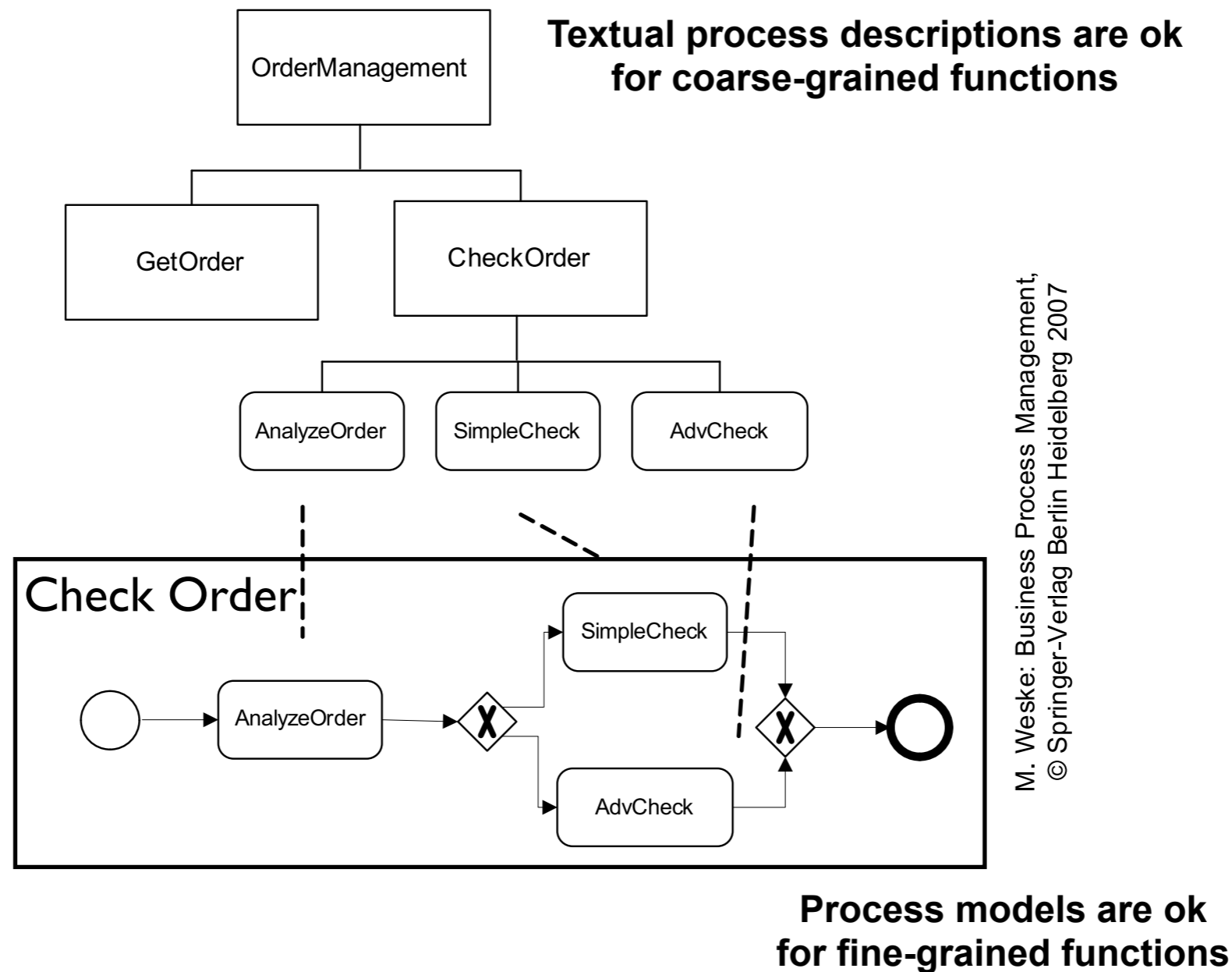


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

Activity models and activity instances



Structuring business processes



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

Vertical Abstraction (separation of concerns)

Guiding principle

Separation of Concerns (SoC)

(to separate a system into distinct features that overlap in functionality as little as possible)

E. W. Dijkstra Archive

the manuscripts of

Edsger W. Dijkstra

1930–2002

Search transcriptions:

Search

[Advanced search.](#)

[Home](#)

[Search](#)

Numerical
EWD Index:

[00xx](#)

[01xx](#)

[02xx](#)

[03xx](#)

[04xx](#)

[05xx](#)

[06xx](#)

[07xx](#)

[08xx](#)

[09xx](#)

[10xx](#)

[11xx](#)

[12xx](#)

[13xx](#)

**Look for
EWD447:
On the role
of scientific
thought**



©2002 Hamilton Richards)

(photo

Edsger Wybe Dijkstra was one of the most influential members of computing science's founding generation. Among the domains in which his scientific contributions are fundamental are

- algorithm design
- programming languages
- program design
- operating systems
- distributed processing
- formal specification and verification
- design of mathematical arguments

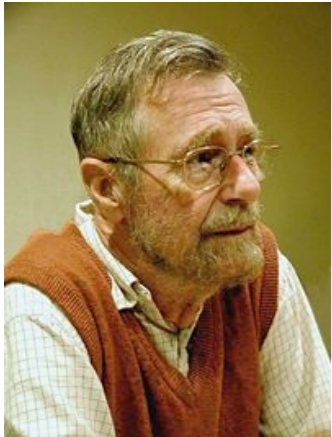
In addition, Dijkstra was intensely interested in teaching, and in the relationships between academic computing science and the software industry.

<https://www.cs.utexas.edu/~EWD/>

(EWD447)

*Let me try to explain to you, what to my taste is **characteristic for all intelligent thinking.***

*It is, that one is willing to **study in depth an aspect of one's subject matter in isolation** for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects.*



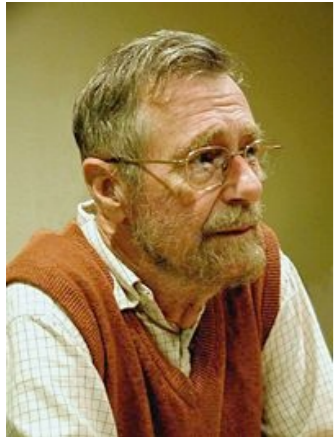
*We know that a program must be **correct** and we can study it from that viewpoint only;*

*we also know that it should be **efficient** and we can study its efficiency on another day, so to speak.*

*In another mood we may ask ourselves whether, and if so: why, the program is **desirable.***

But nothing is gained —on the contrary!— by tackling these various aspects simultaneously.

(EWD447)



*It is what I sometimes have called **the separation of concerns**, which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of.*

It does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant.

Business data processing systems are sufficiently complicated to require such a separation of concerns.

*The suggestion that in that part of the computing world "**scientific thought is a non-applicable luxury**" puts the cart before the horse: the mess they are in has been caused by **too much unscientific thought.***

(1974)

SoC: an example

HyperText Markup Language (HTML):
organization of webpage content

Cascading Style Sheets (CSS):
definition of content presentation style



JavaScript (JS):
user interactions

From your forms

6. Are you familiar with HTML, CSS and XML?

[Altri dettagli](#)

 Dati analitici

 Yes	17
 Just a little	16
 No	26



```
<!DOCTYPE html>
<html>
<style>
body {
  background-color: lightblue;
}
h1 {
  color: darkblue;
  text-align: center;
}
p {
  font-family: verdana;
  font-size: 20px;
}
</style>

<body>

<h1>HTML, CSS and JAVASCRIPT</h1>

<button type="button"
  onclick="document.getElementById('demo').innerHTML = Date()";
Click me to display Date and Time.
</button>

<p id="demo"></p>

</body>
</html>
```

CSS

Javascript



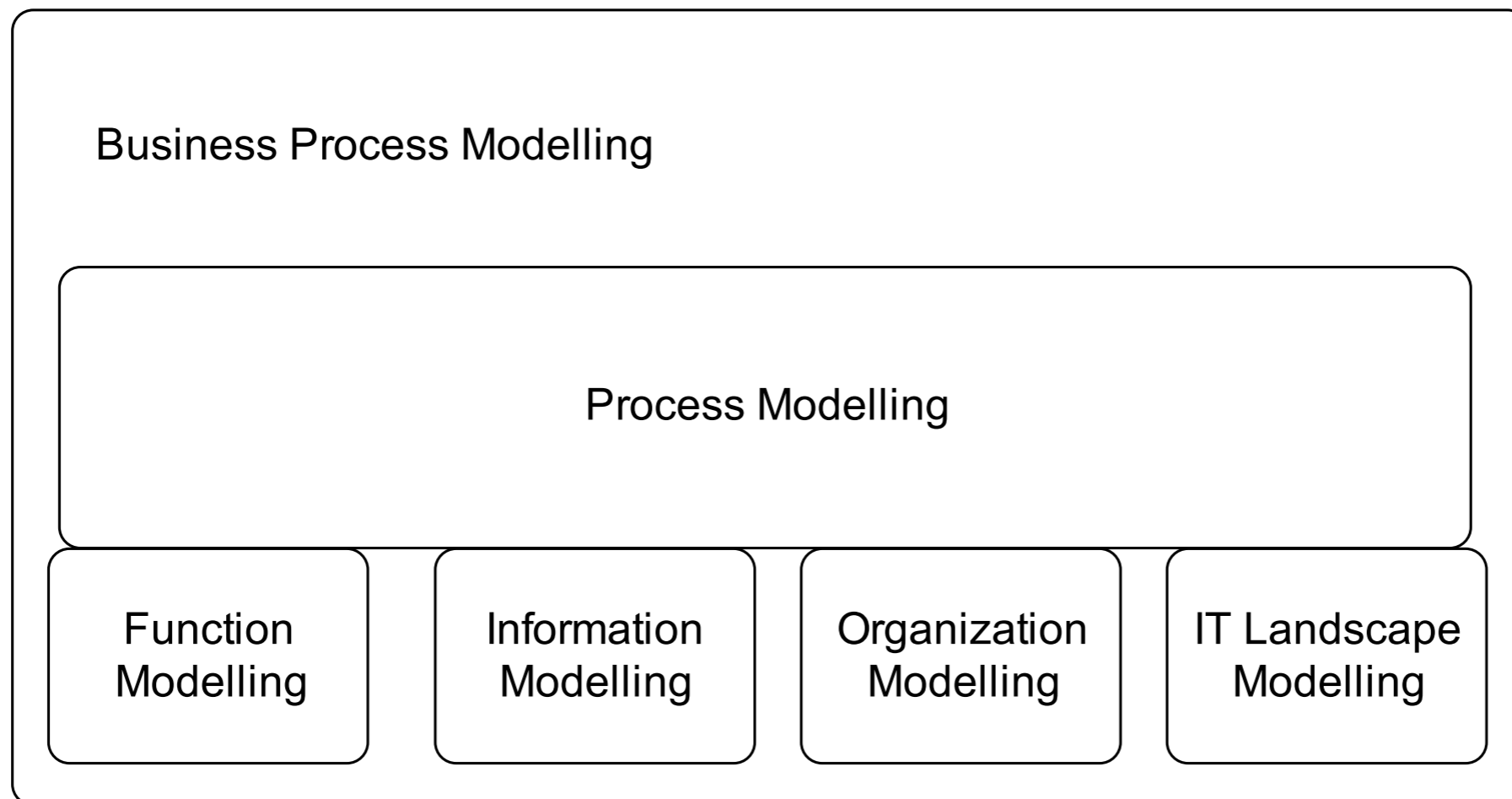
```
<style>
body {
  background-color: lightyellow;
}
h1 {
  color: darkred;
  text-align: center;
}
p {
  font-family: courier;
  font-size: 15px;
}
</style>
```

CSS



Vertical abstraction (domain separation)

BPM includes multiple modelling domains,
integrated by Process Modelling

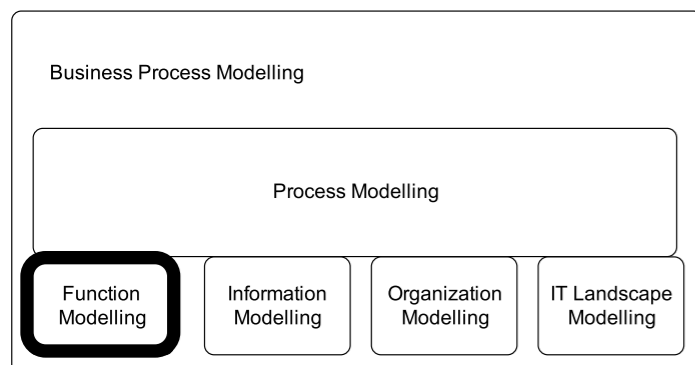


Function models

Units of work enacted by processes
(at different levels of granularity)

Informal description, textual documents
(coarse-grain business level)

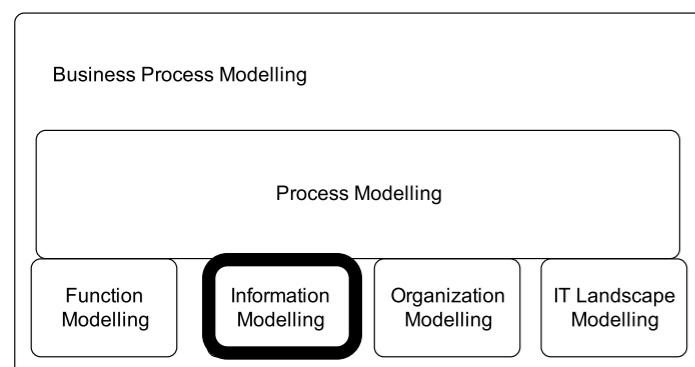
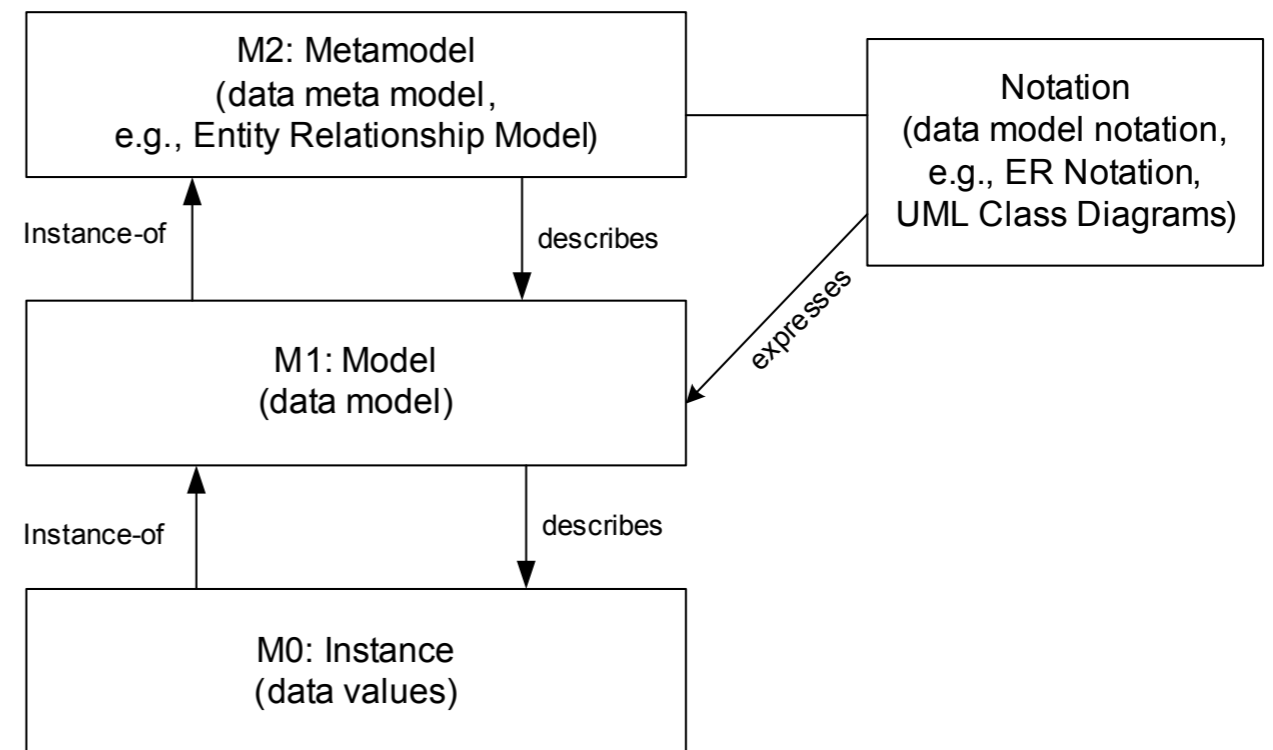
Formal description, function specifications
(fine-grain software layer)



Information models

Data representation is crucial:
all decisions made during a
business process depends on
data values

Data dependencies between
activities are also important
(ensure data-availability,
reduce waiting time)



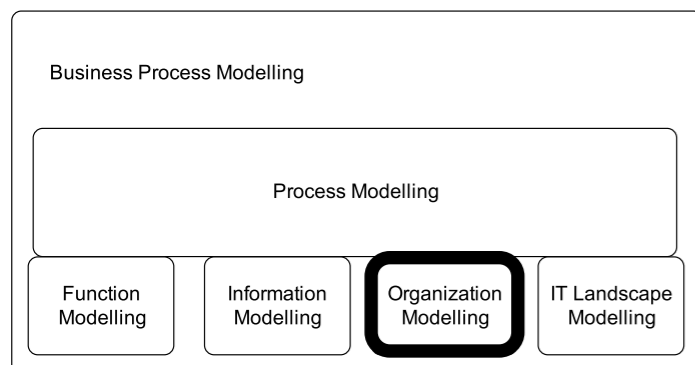
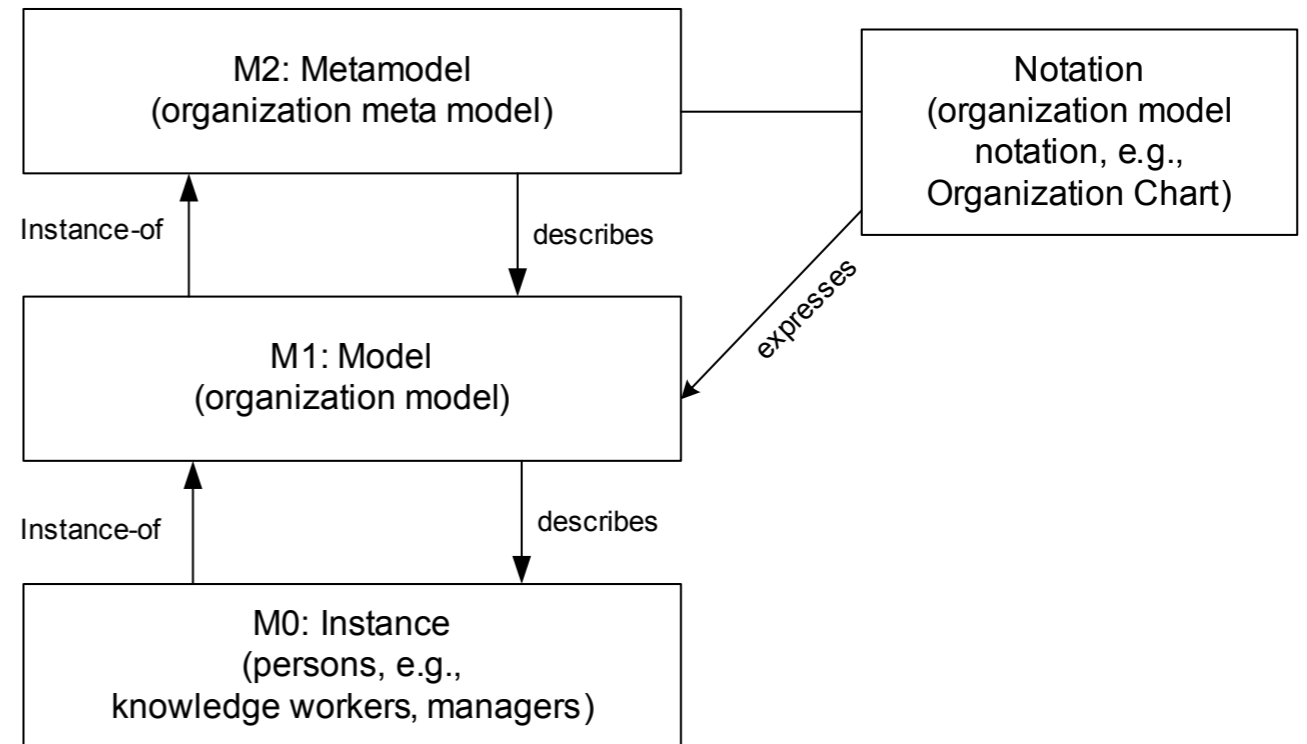
Organizational models

Organizational structure must also be represented

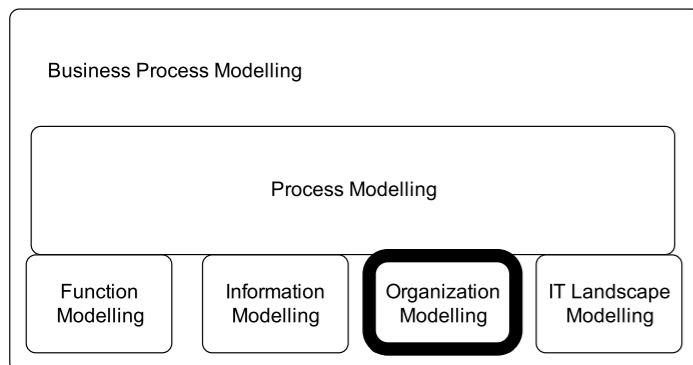
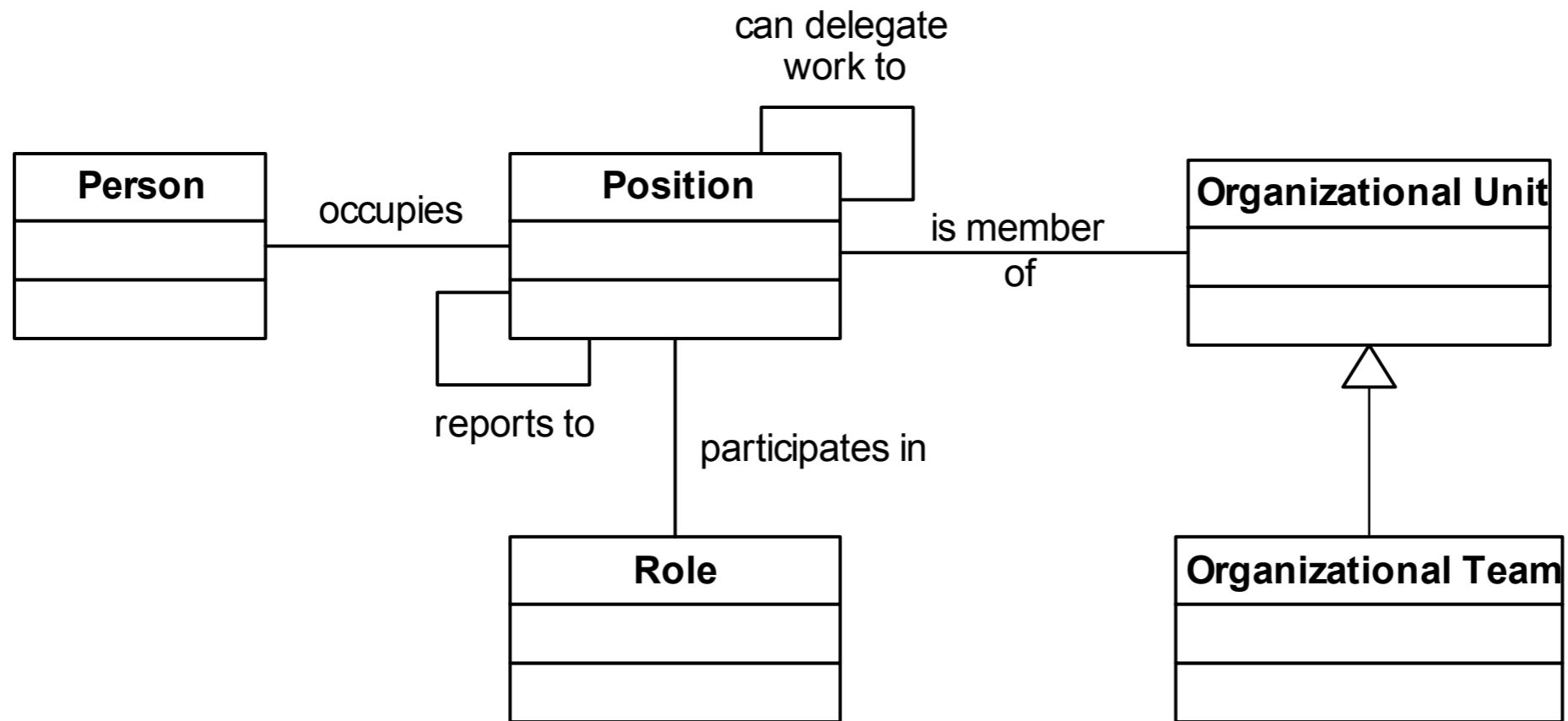
Activities can be associated to specific roles or depts

Roles are groups of employees that qualify for being responsible of certain activities

Increased flexibility:
different persons can cover the same role at different time in different cases



An organizational metamodel

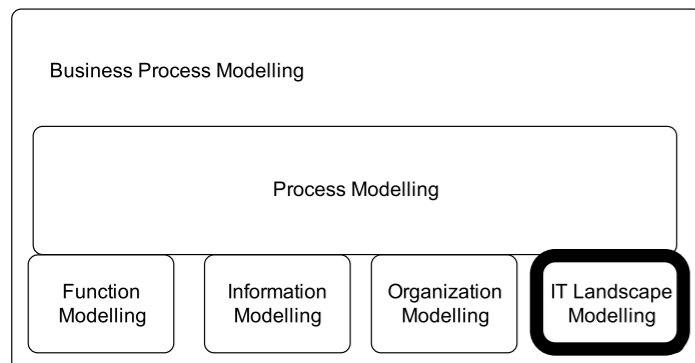


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

IT landscape

Many activities in a business process are supported by information systems

Information systems and programming interfaces needs to be represented because they provide functionalities



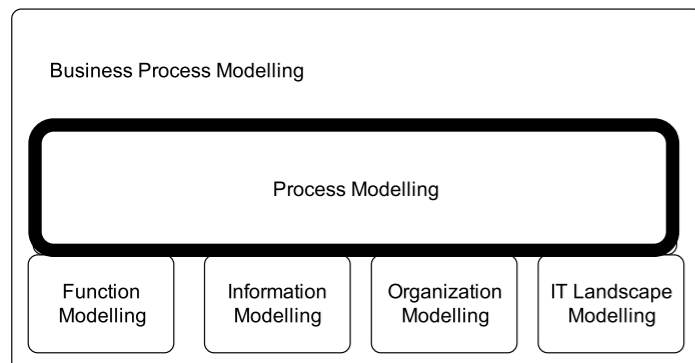
Process models

Define the glue between the subdomains

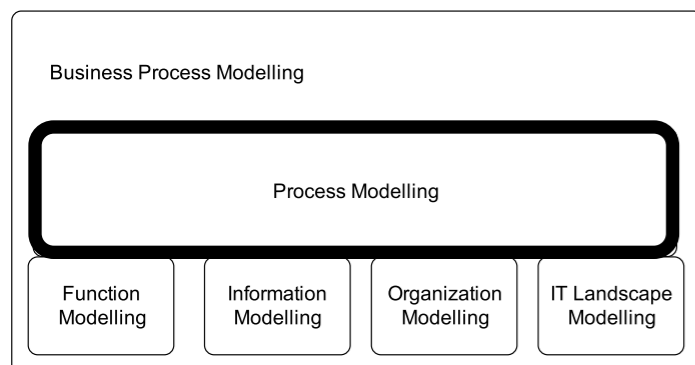
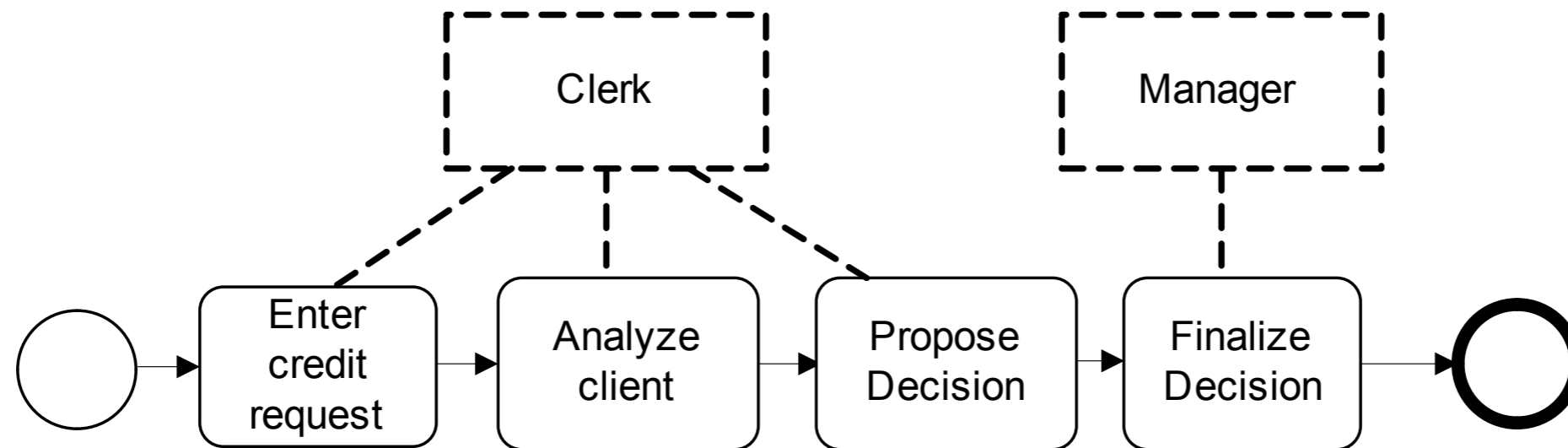
Relate activities and execution constraints

Relate data values with process instances
(e.g. the process of a credit approval may depend on the requested amount)

Assign activities to roles

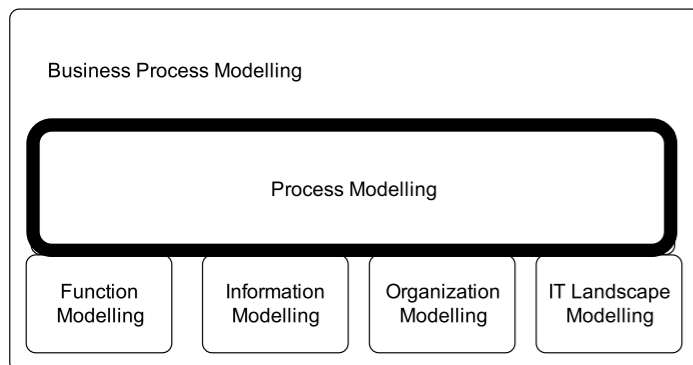
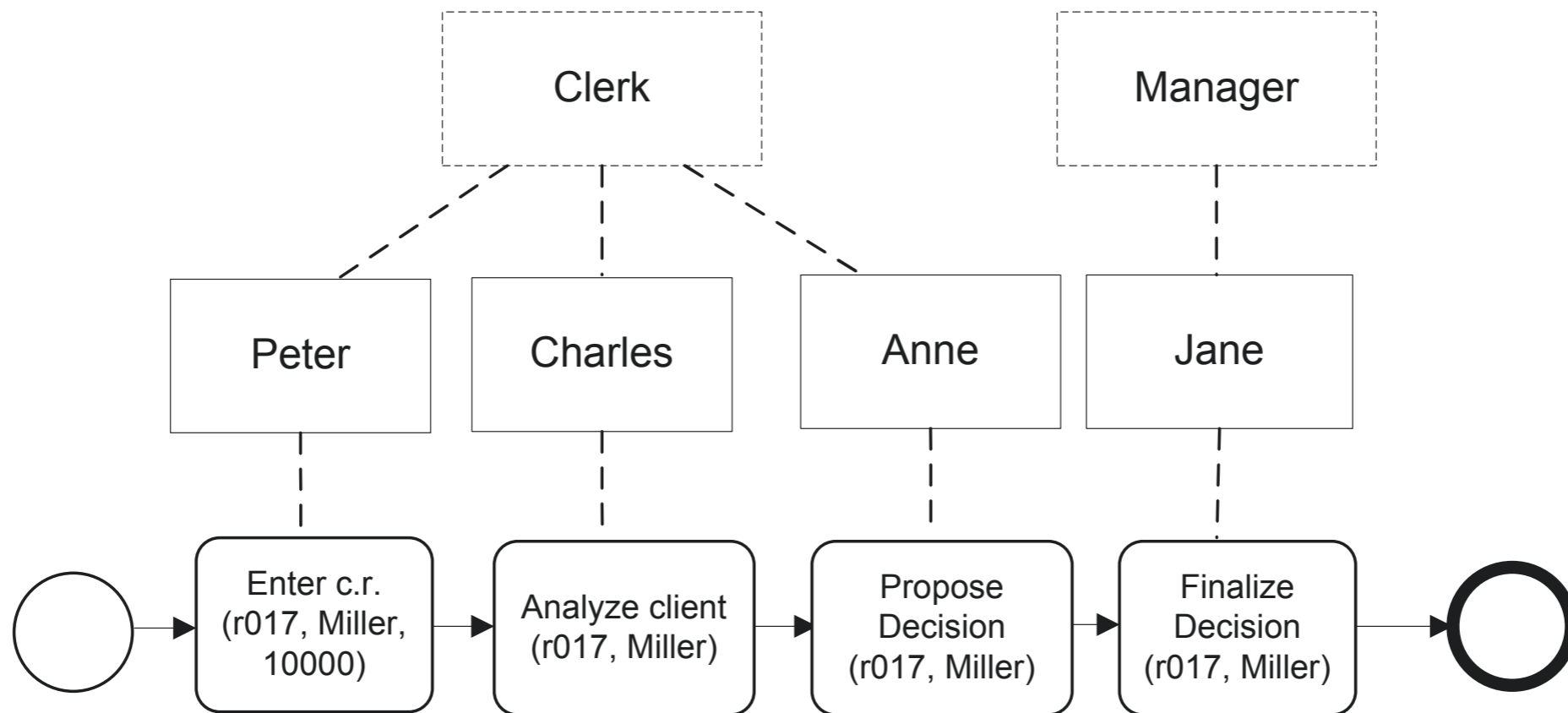


A process model with role information



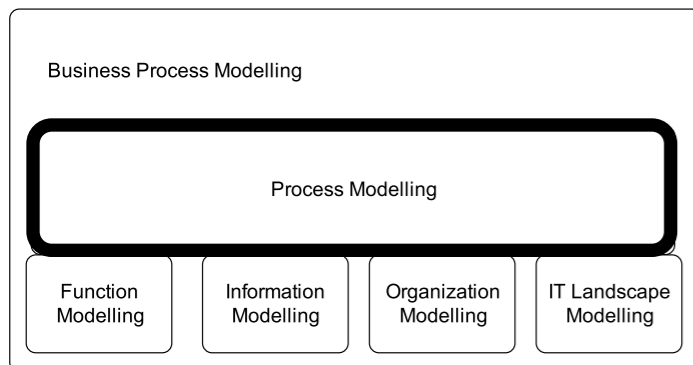
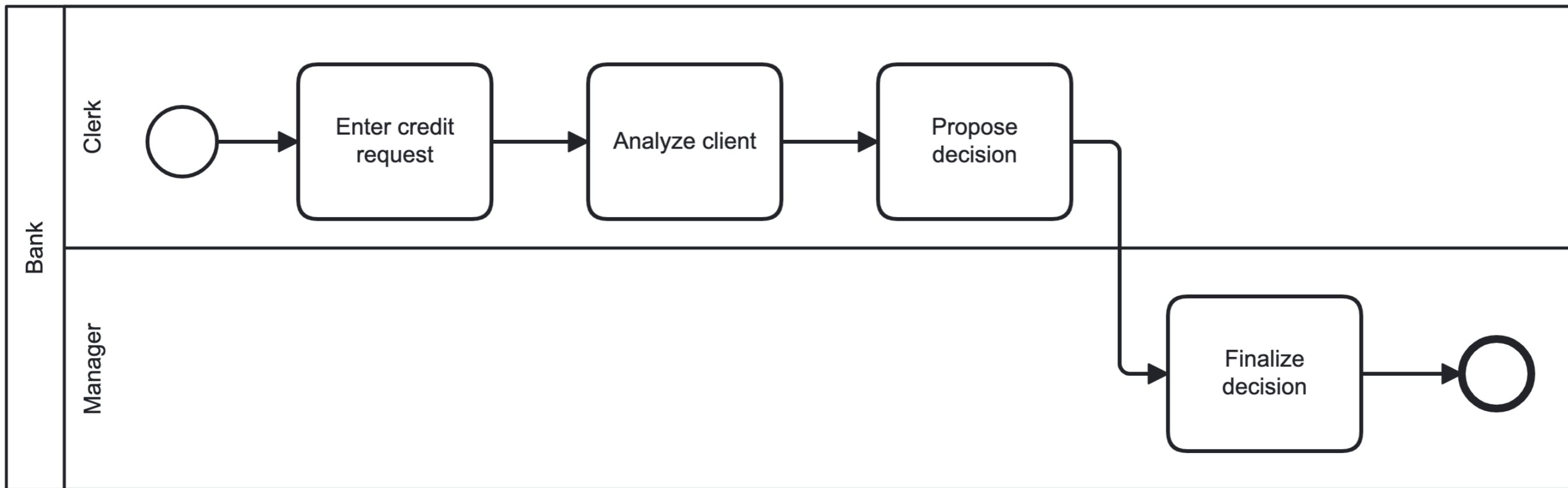
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

A process instance with role information



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

A BPMN model with role information



A BPMN model with data objects

