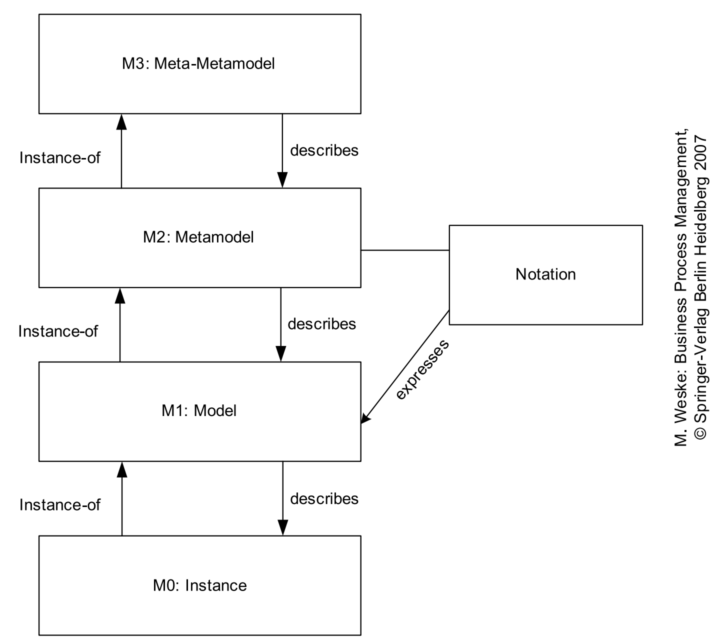
Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni
http://www.di.unipi.it/~bruni

04 - Models and Abstraction

Object



Overview of the conceptual models
and abstraction mechanisms
in business process modeling

Ch.1 of Workflow Management: Models, Methods, and Systems
Ch.1 of Business Process Management: Concepts, Languages, Architectures
Ch.3.1--3.3 of Business Process Management: Concepts, Languages, Architectures

Some definitions

Workflow management coalition (WfMC)

Founded in the ‘90s by vendors, users, academia: fix standard for Wf representation and execution

http://www.wfmc.org

Workflow

Definition: a **workflow** is the automation of a business process, in whole or in part,

during which documents, information, or tasks are passed from one participant to another for action,

according to a set of procedural rules

Workflow Management Coalition

WfMC

Workflow management system

Definition: a **workflow management system**
is a software system
that defines, creates, and manages Wfs execution,
running on one or more workflow engines,
able to interpret the workflow definition,
able to interact with workflow participants, and
able to invoke the use of IT tools and applications

Kinds of workflow

Definition: a **system workflow** consists of activities that are implemented by software systems without any user involvement

Definition: Workflows in which humans are actively involved and interact with information systems are called **human interaction workflows**.

Human interaction workflows

Goal:

support automation by driving the human activities according to the process model

Benefits:

reduce idle periods

avoid redundant work

improve human/machine work integration

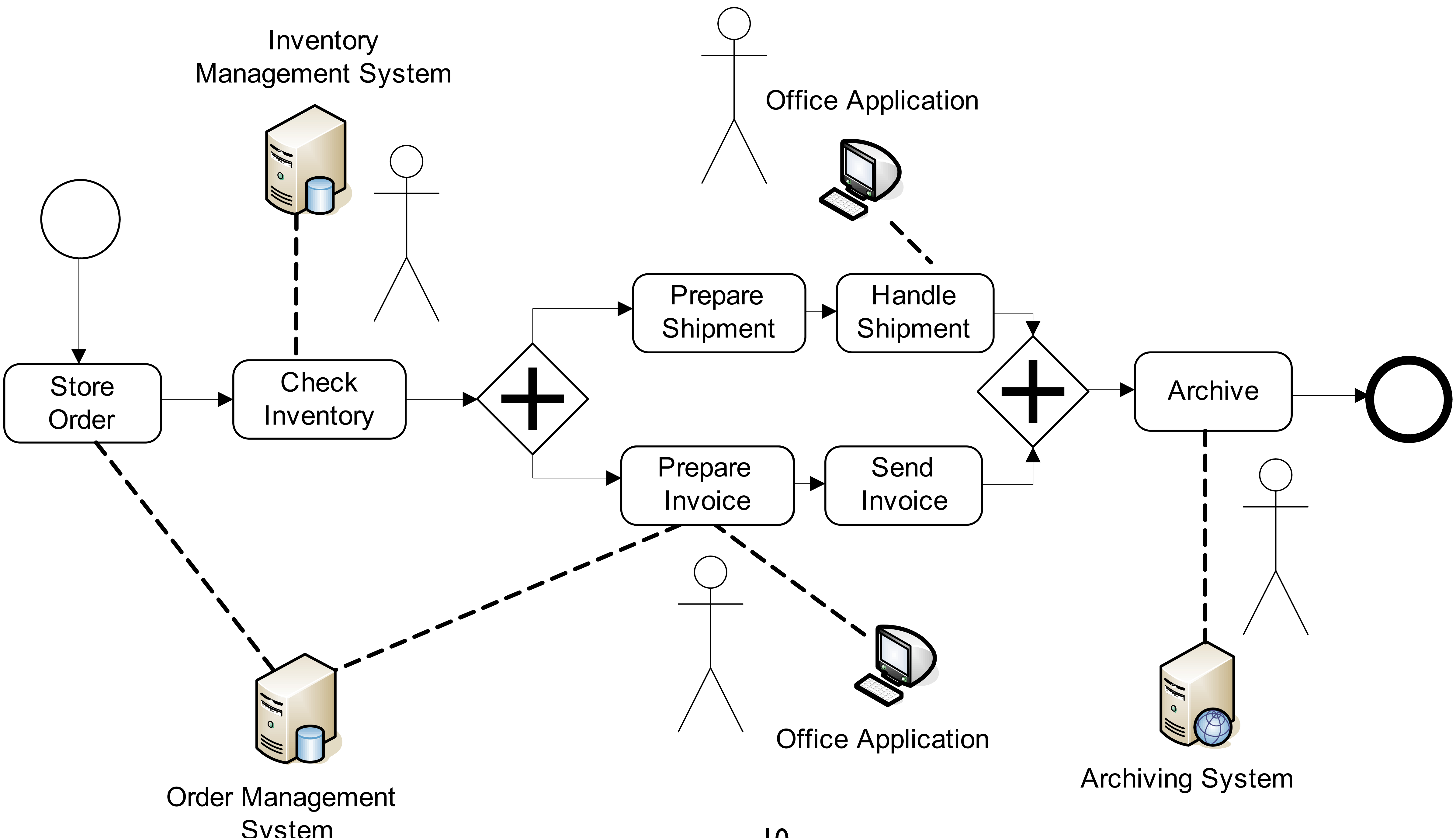
Human collaboration

When task performed by humans are involved in the workflow, it is not sufficient to equip workers with adequate software:

their collaboration must be supported

shared data repositories and work handover can speed-up office procedure considerably

Example: Human interaction workflow

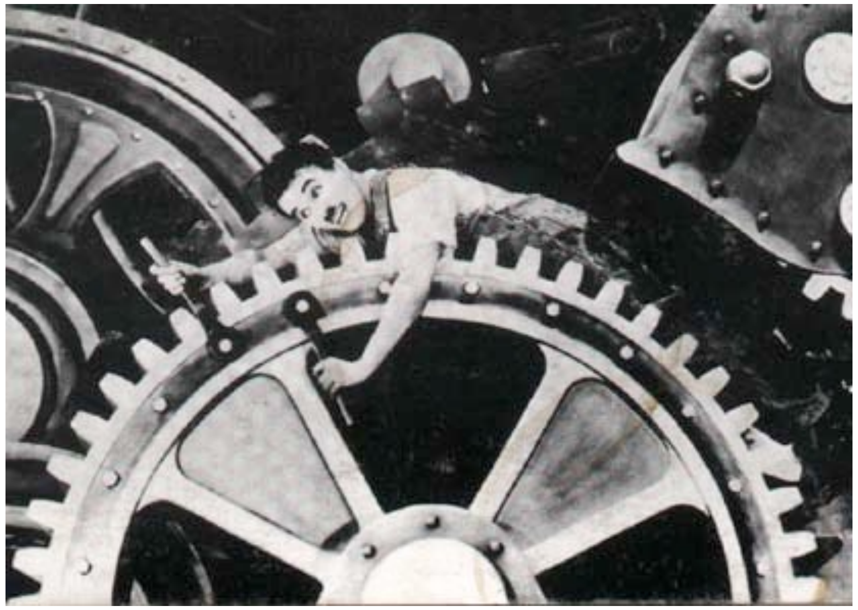


Some limitations

Problems with knowledge workers:

User acceptance issues

Machine burdening of workers



Little room for creativity and flexibility

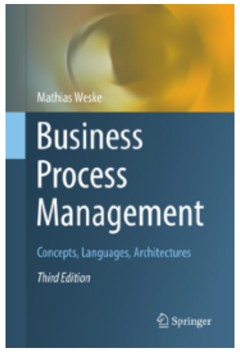
Business process

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.

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

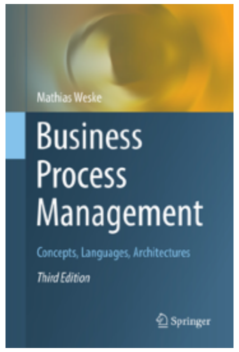
- *Weske*



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



Business process management

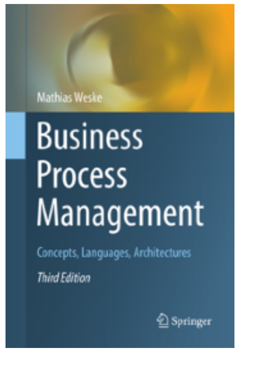
The basis of business process management is the **explicit representation** of business processes with their **activities** and the **execution constraints** between them

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

Business process management system

Definition: **business process management system** is a generic software system that is driven by explicit process representations to coordinate the enactment of business processes.

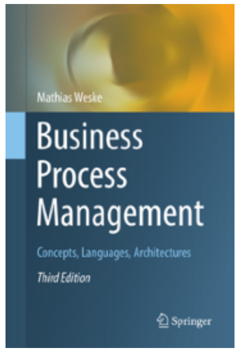
- Weske



Business process model

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

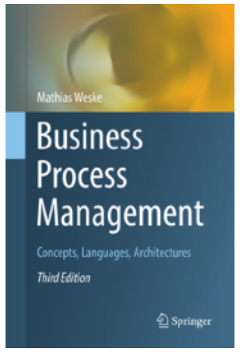
- Weske



Business process instance

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

- Weske



Model and instances

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
as well as activity instances

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

Process-driven software

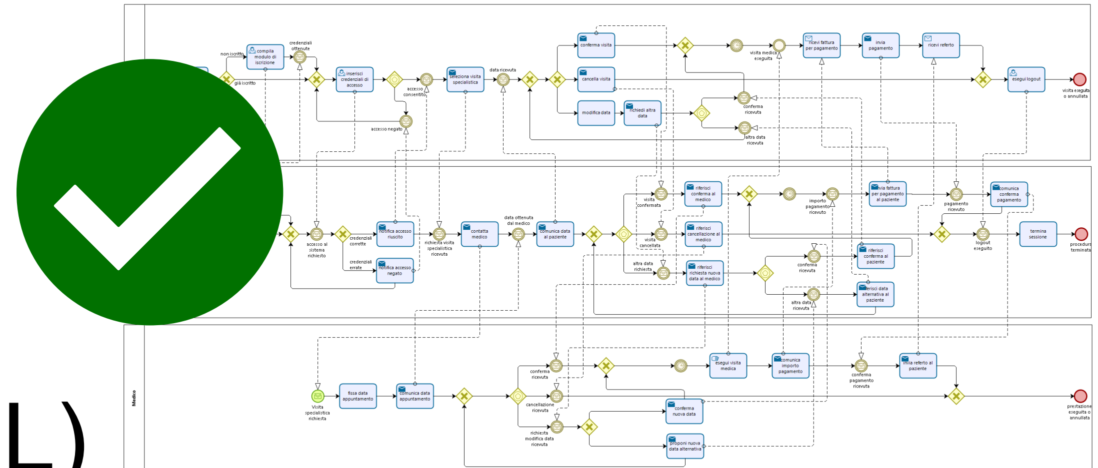
Business process models are the main artifact for implementing business processes

This implementation can be done by organizational rules and policies, but it can also be done by business process management (software) system

In this case the software system is driven by explicit process representations (models)

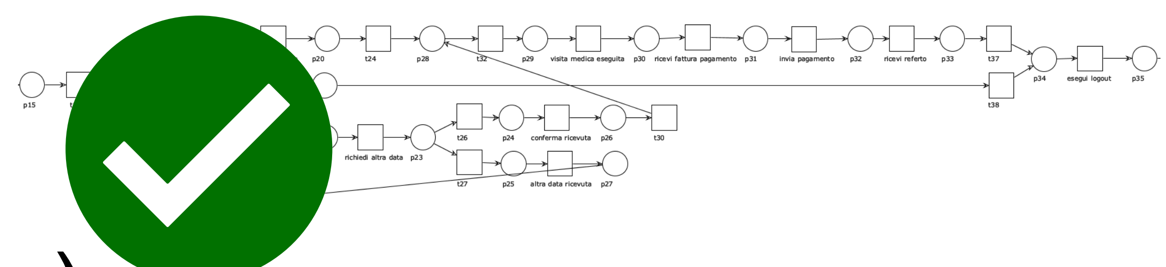
Process representations

Visual representations:
diagrams and charts
understandable by humans
(informal, intuitive, BPMN, EPC, BPEL)



Languages:
unambiguous machine syntax
(process dialects, XML schemes)

Models:
rigorous semantics for scientists
(automata, Petri nets, workflow nets)



Models and abstraction

Models

A model is a simplified representation of reality

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

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

A way to cope with complexity

Abstractions

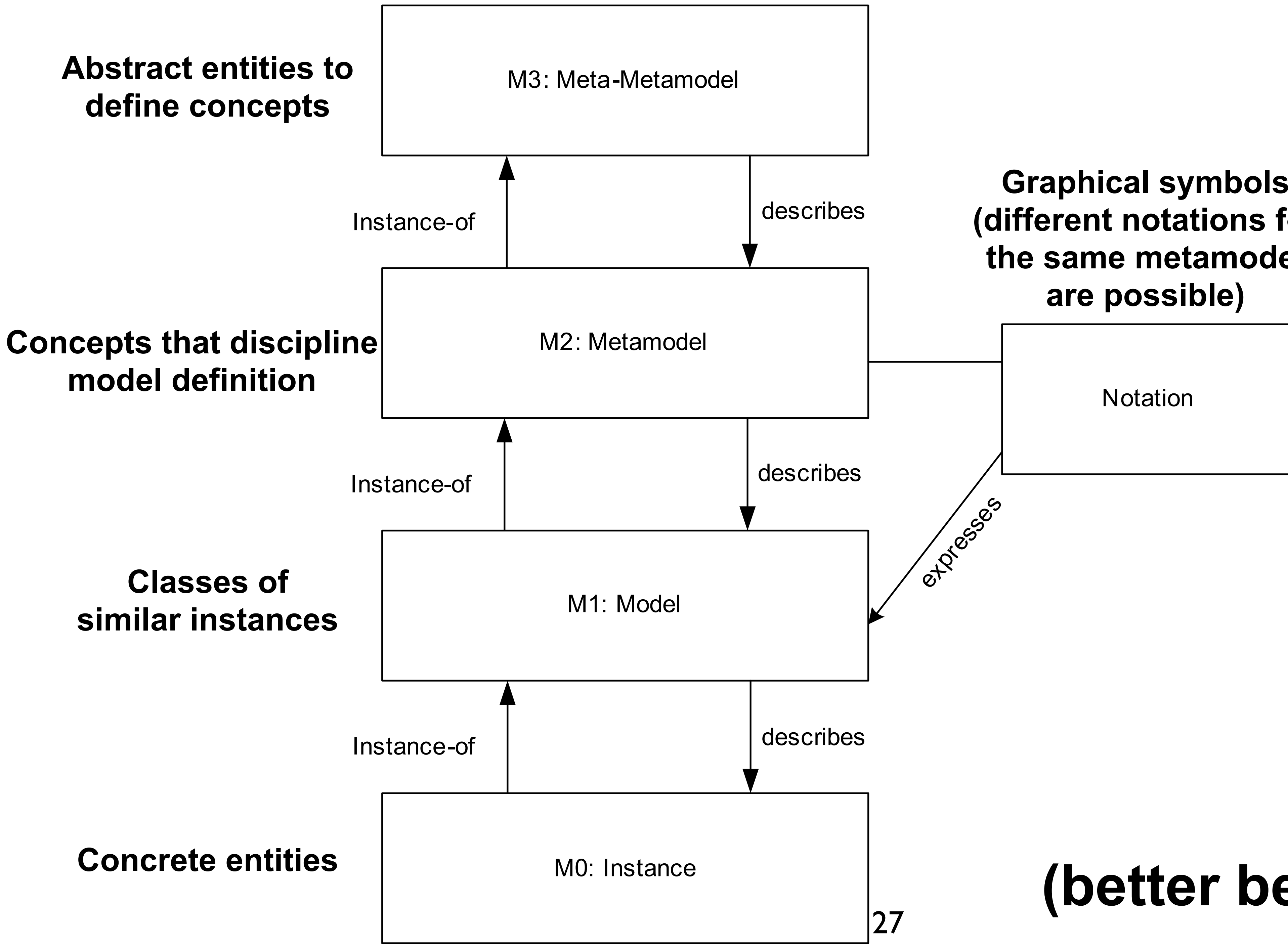
Horizontal: separation at different modeling levels

Aggregation: separation at different granularity levels

Vertical: separation at different subdomains

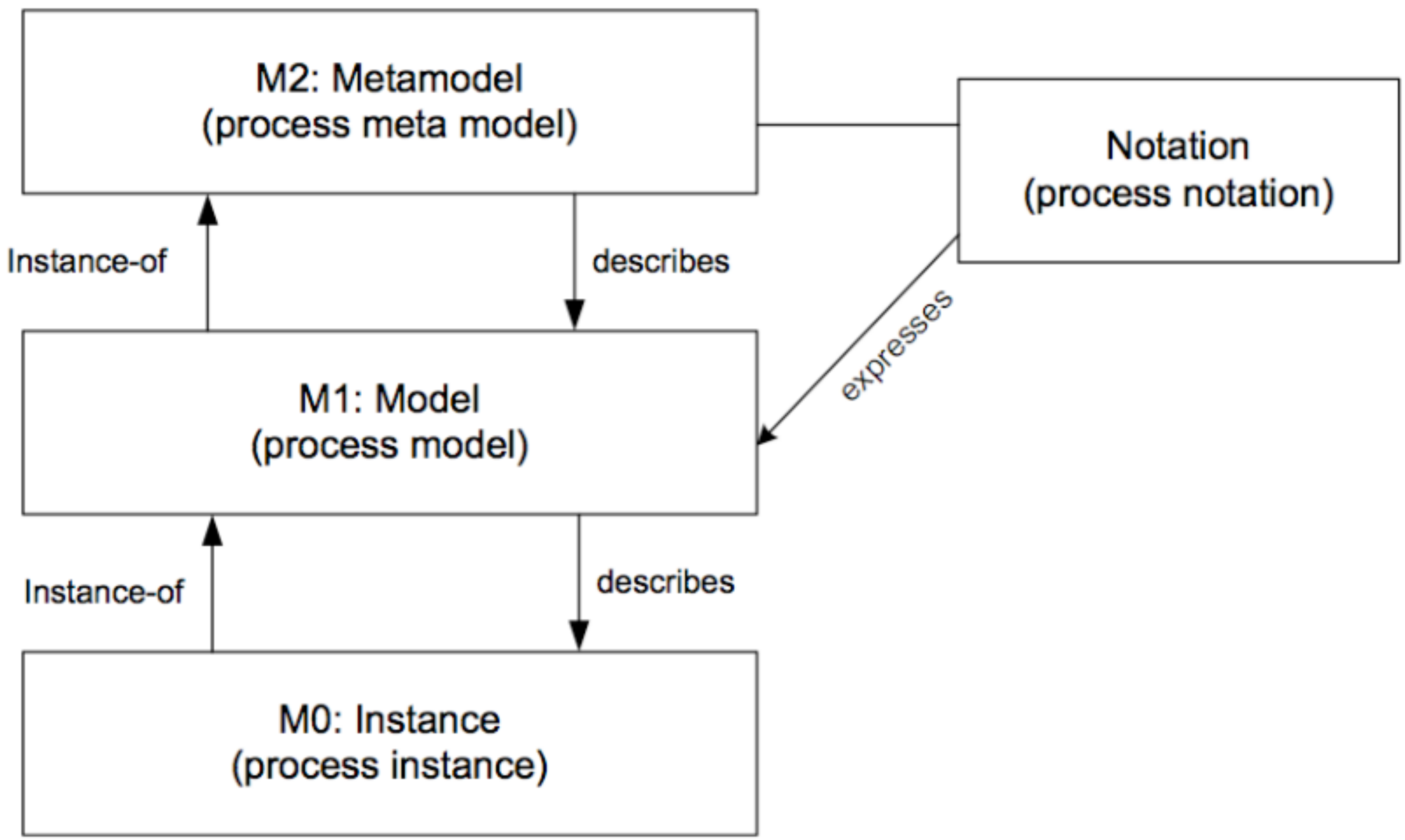
Horizontal Abstraction

Horizontal abstraction (modeling levels)

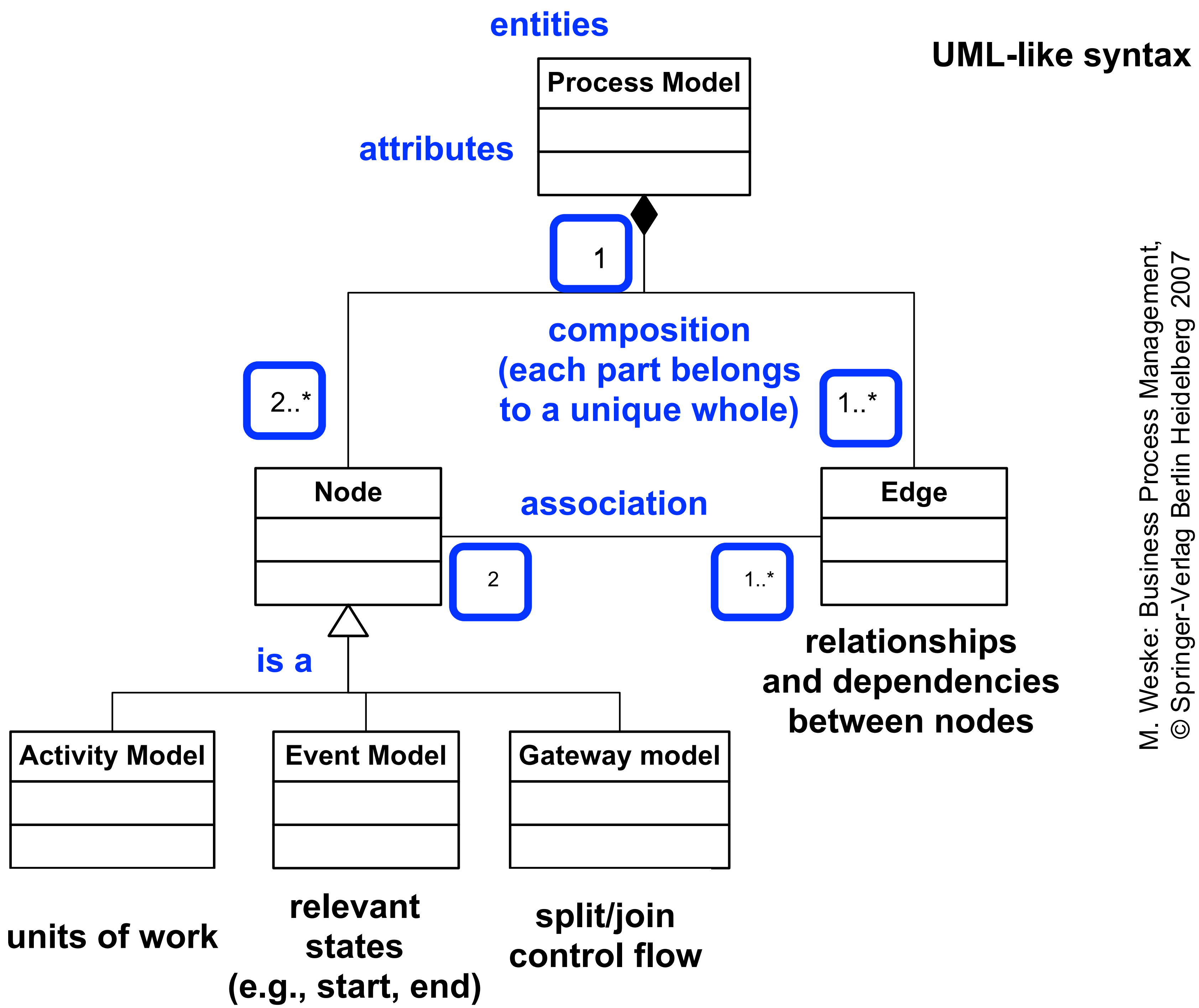


(better be read bottom-up)

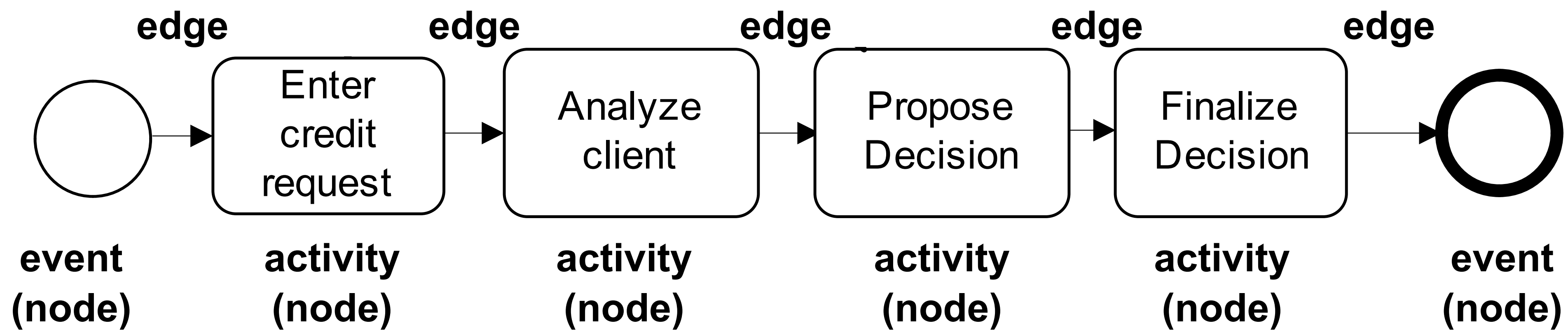
Process models and process instances



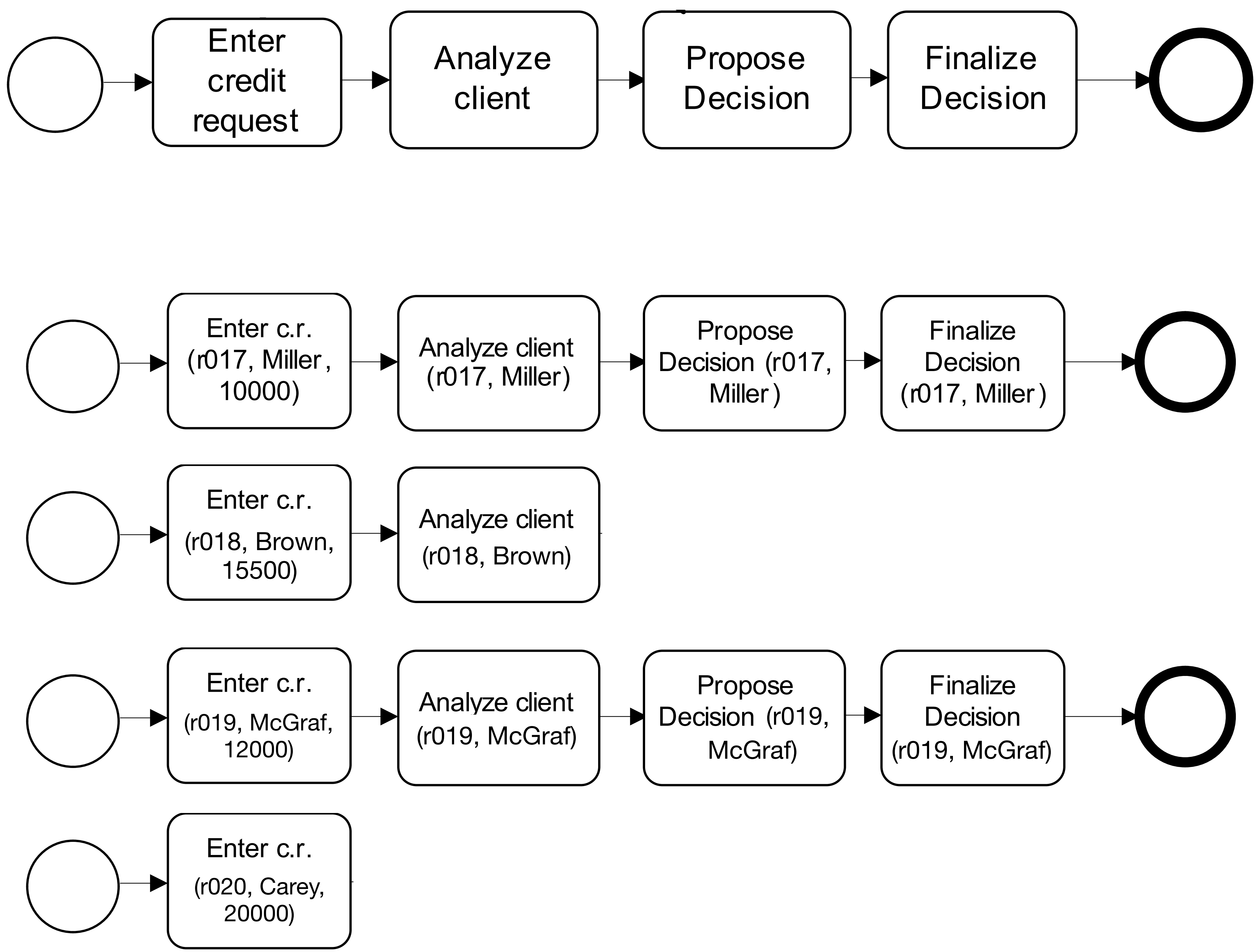
A process metamodel (level M2)



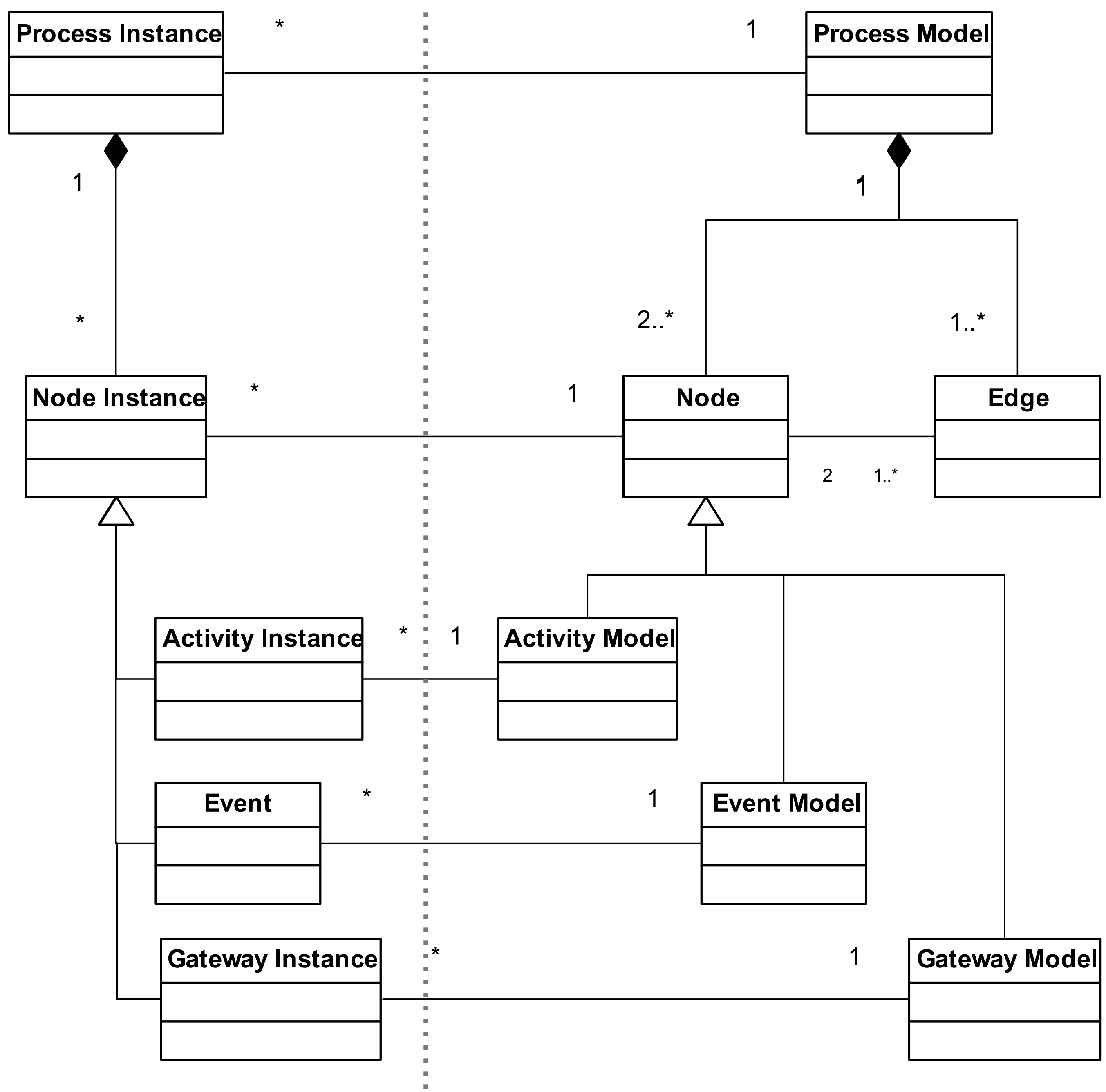
A process model



Some process instances

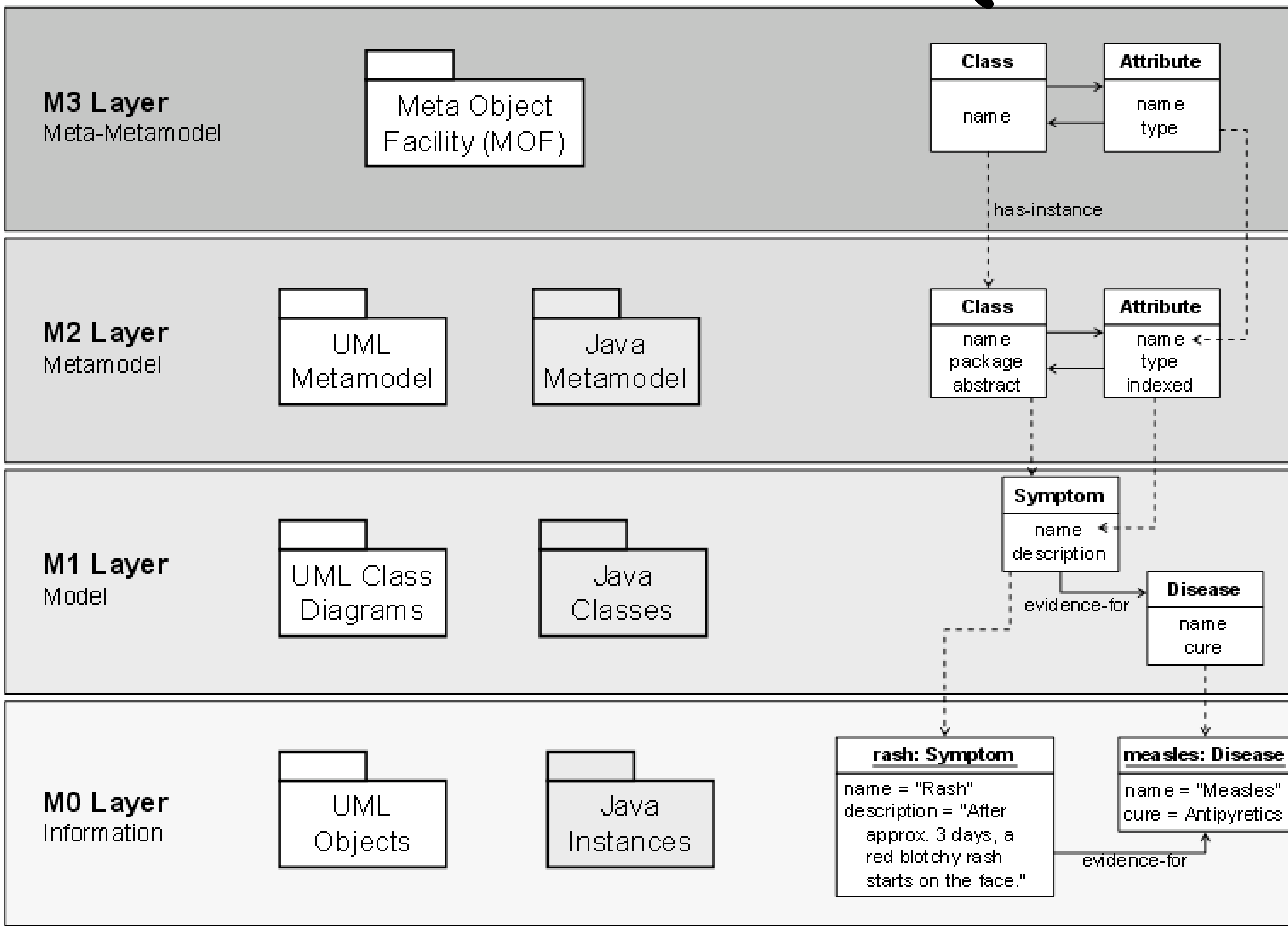


Process models and process instances



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

An example: MOF metamodel (OMG)



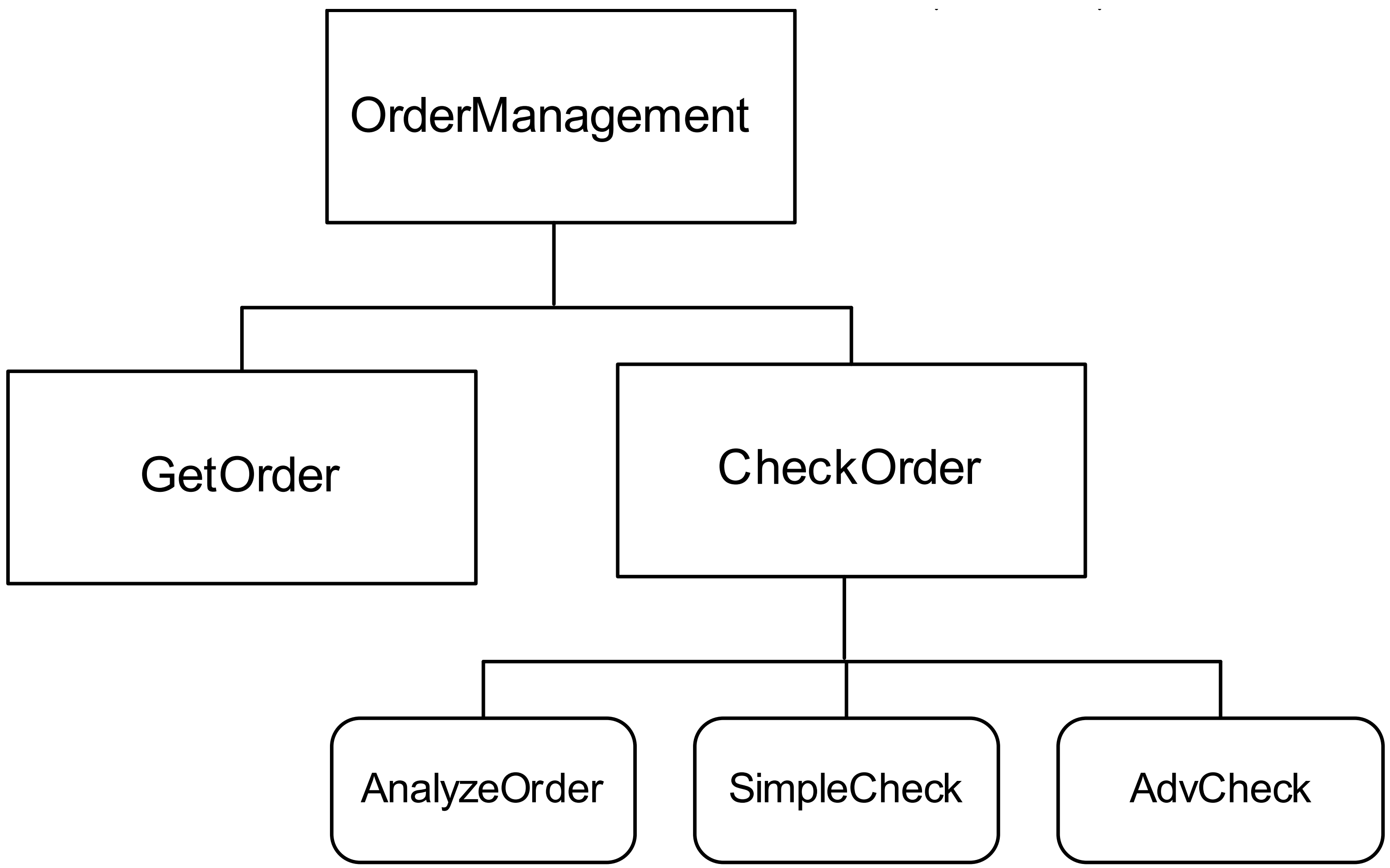
Aggregation Abstraction

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

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

A sample aggregation



Vertical Abstraction

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

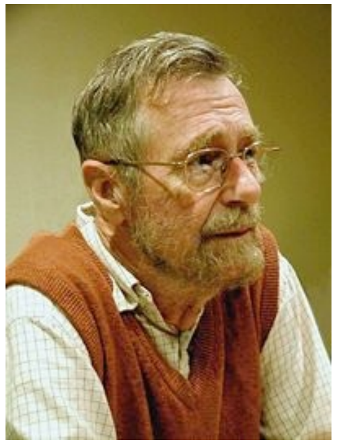
(photo ©2002 *Hamilton Richards*)

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.

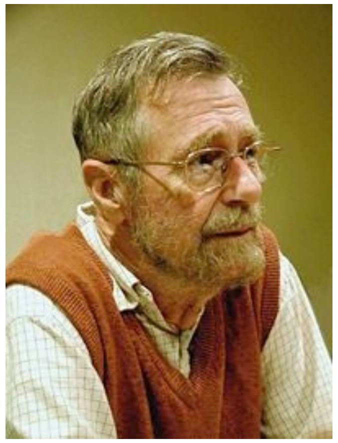
http://www.cs.utexas.edu/users/EWD/



On the role of scientific thought (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.



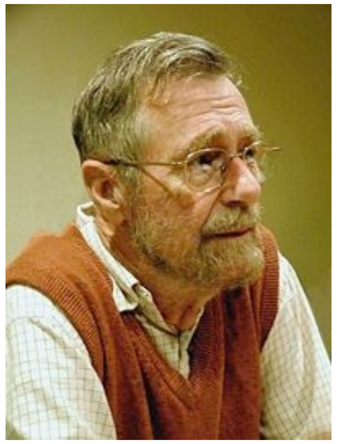
*We know that a program must be **correct** and we can study it from that viewpoint only;*
*we also know that it should be **efficient** and we can study its efficiency on another day, so to speak.*
*In another mood we may ask ourselves whether, and if so: why, the program is **desirable**.*

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

*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.



On the role of scientific thought (EWD447)

… and 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**....*

SoC: an example

HyperText Markup Language (HTML):
organization of webpage content

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

JavaScript (JS):
user interactions

```
<!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

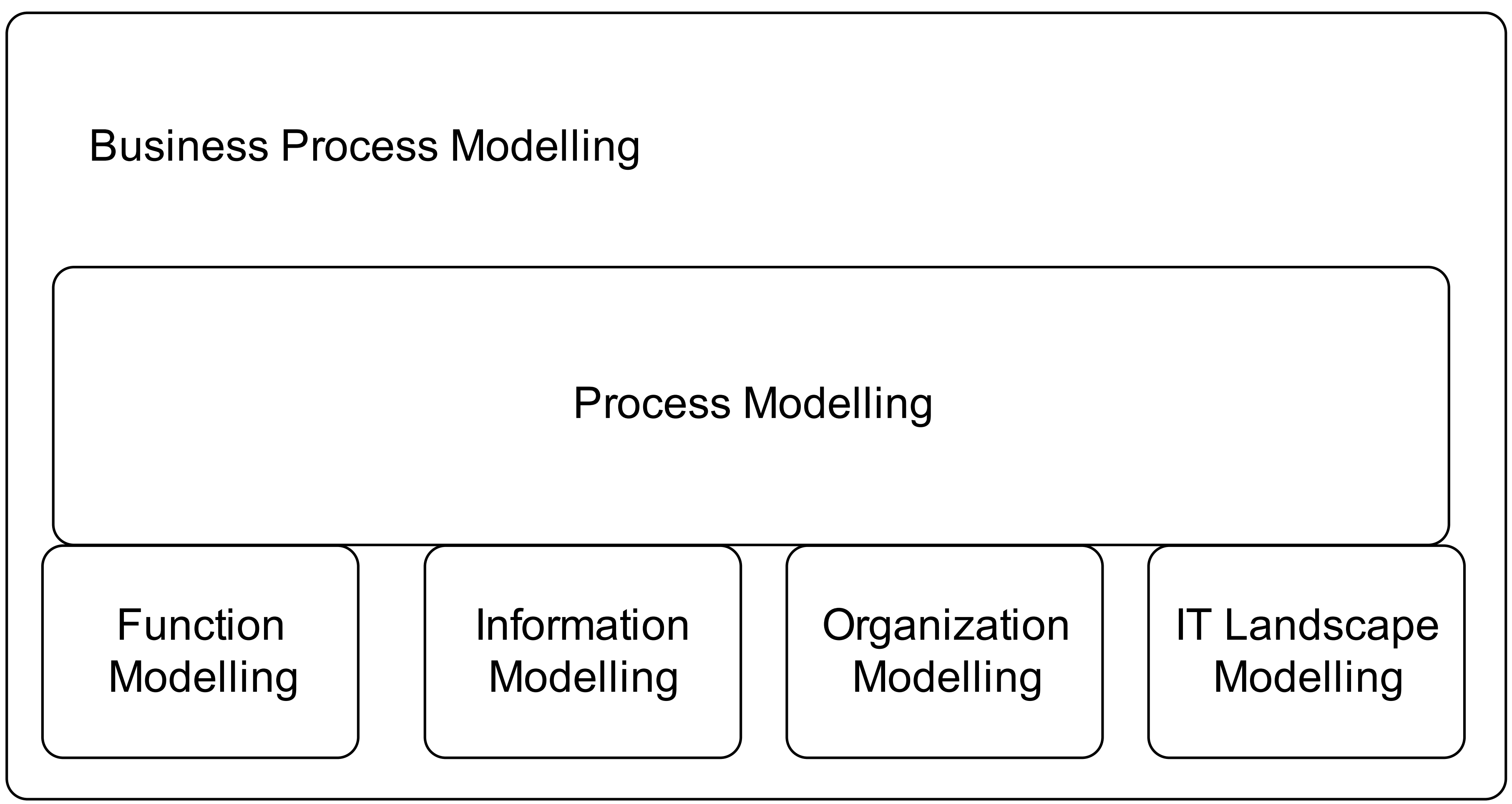
Javascript


```
<!DOCTYPE html>
<html>
<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

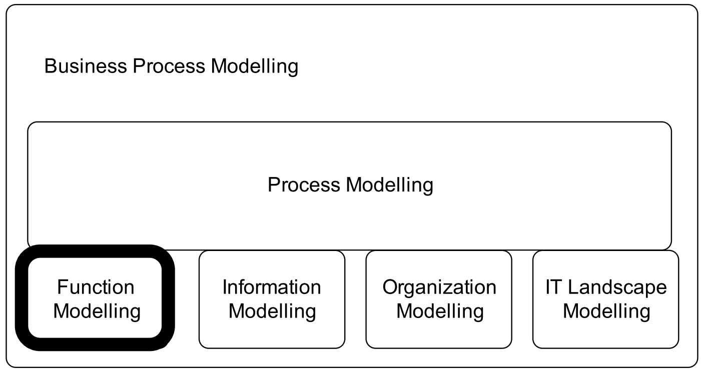


Function models

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)

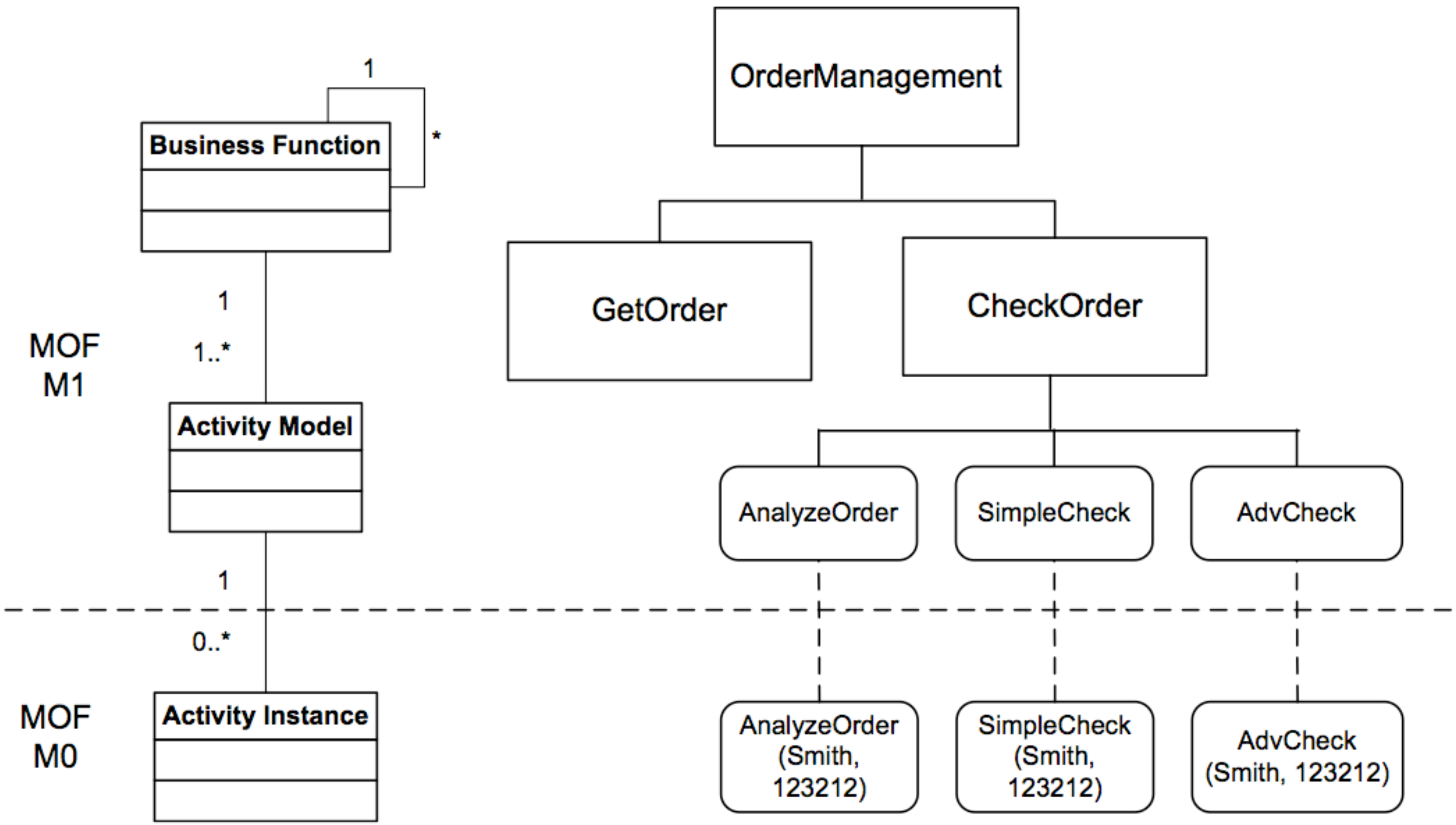


High-level business functions

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)

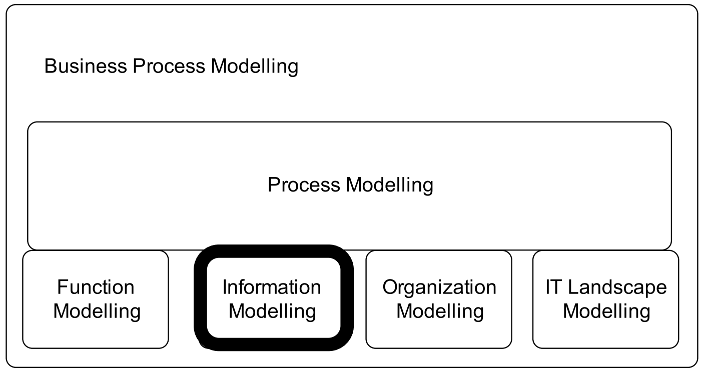
Activity models and activity instances



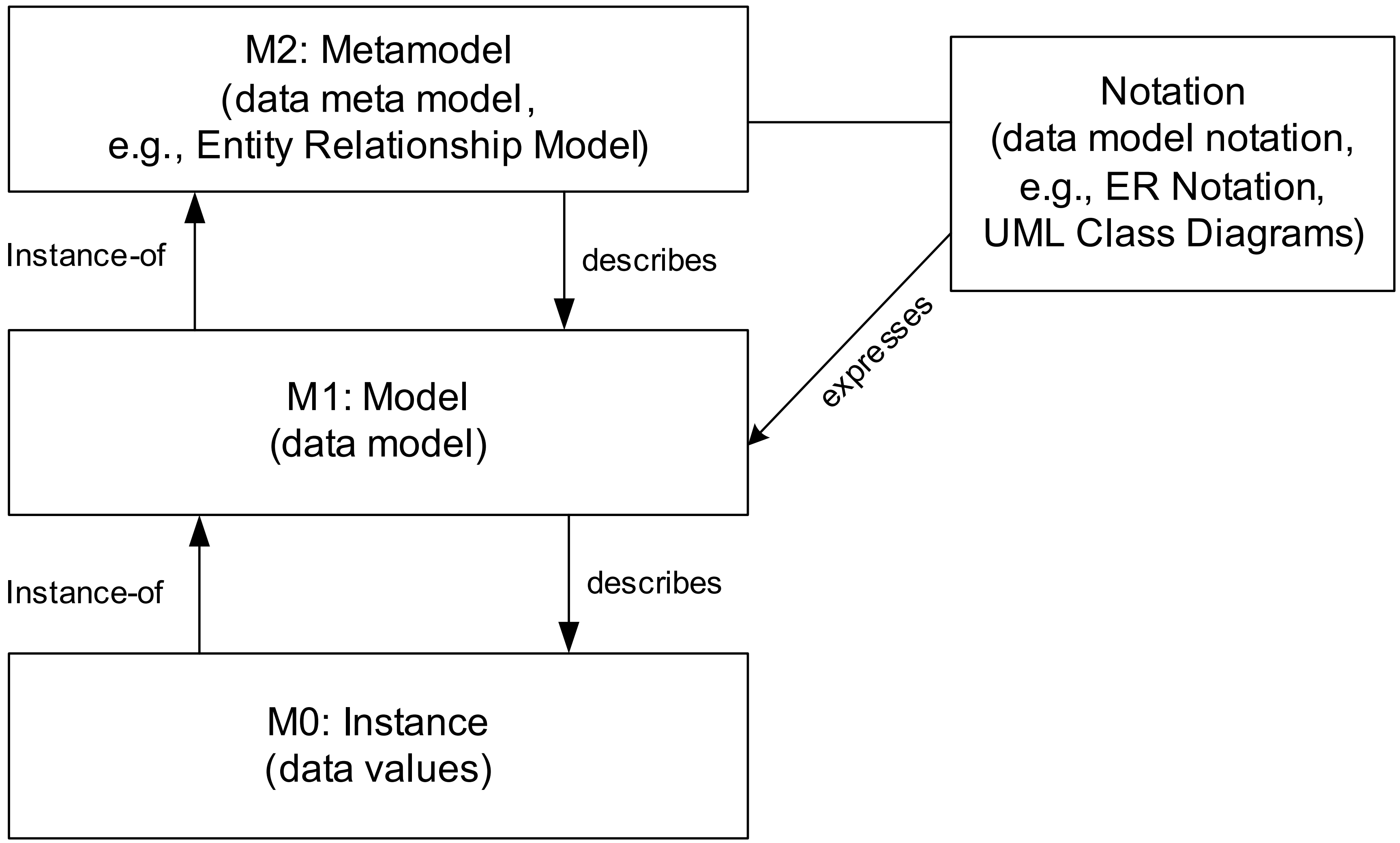
Information models

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

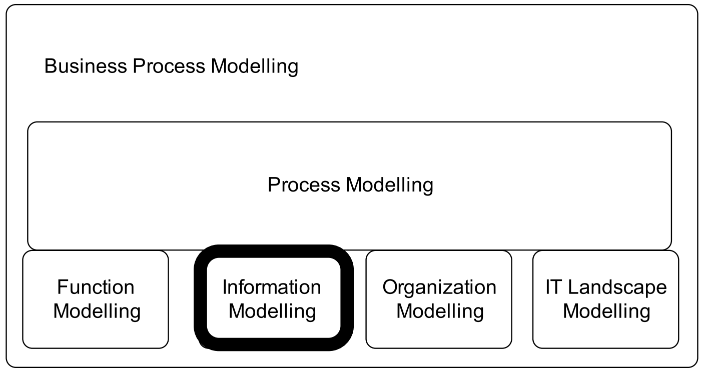
Data dependencies between activities are also important
(ensure data-availability, reduce waiting time)



Data models



M2: Metamodel
(data meta model,
e.g., Entity Relationship Model)
Notation
(data model notation,
e.g., ER Notation,
UML Class Diagrams)
Instance-of
describes
expresses
M1: Model
(data model)
Instance-of
describes
M0: Instance
(data values)

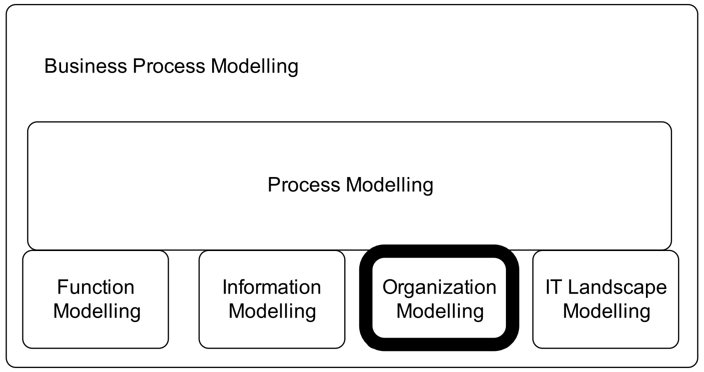


Business Process Modelling
Process Modelling
Function Modelling
Information Modelling
Organization Modelling
IT Landscape Modelling

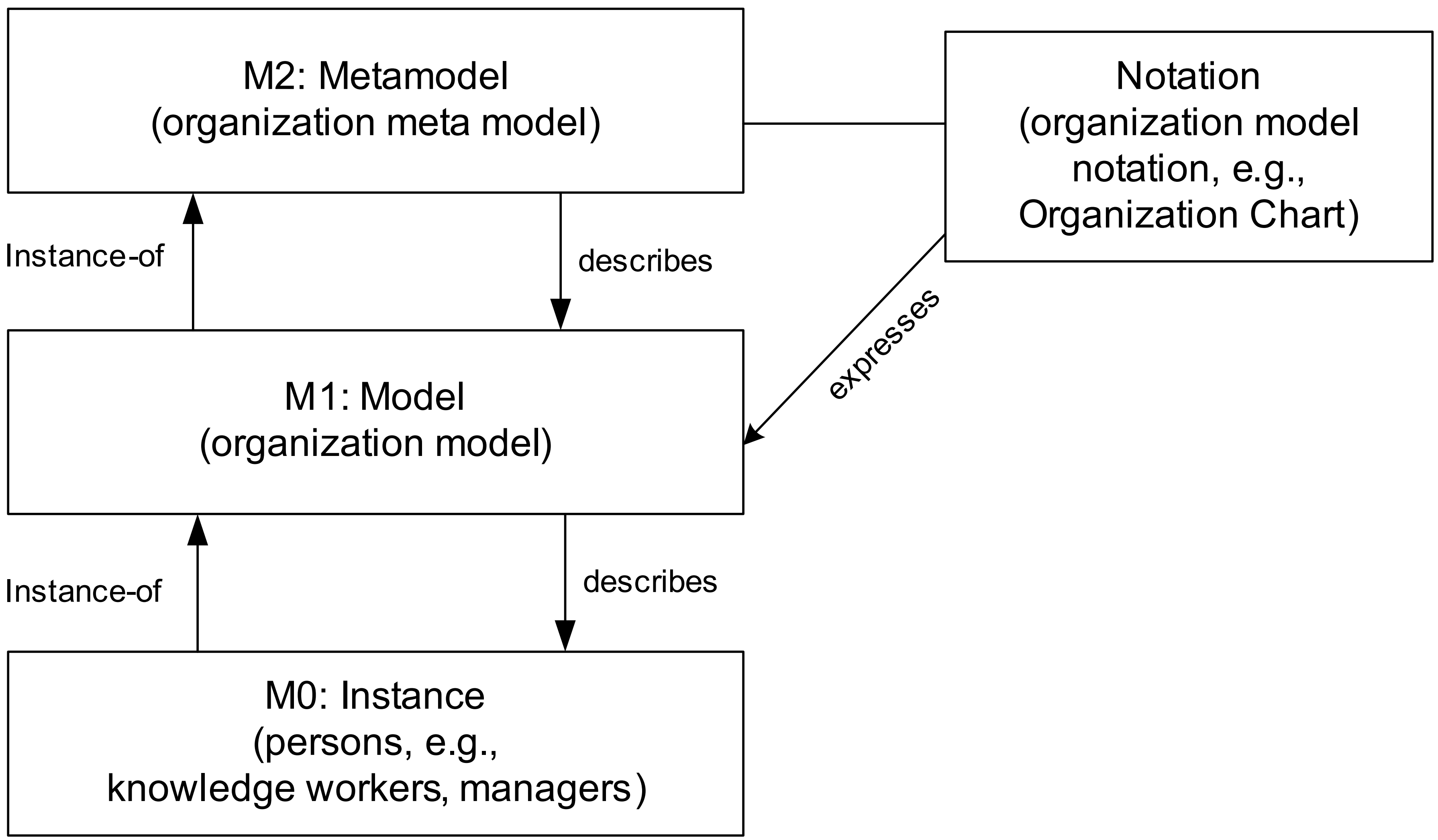
Organizational models

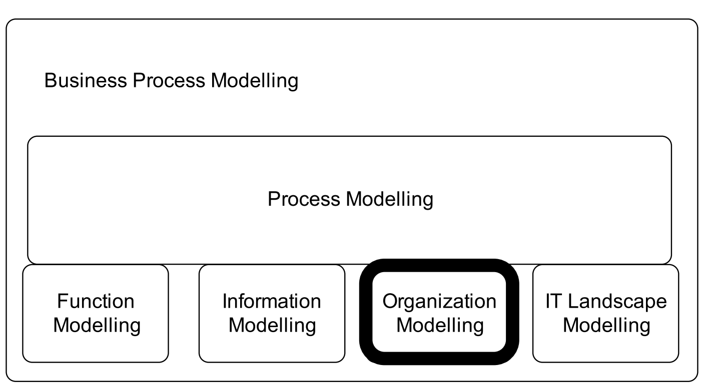
Organizational structure must be represented

Activities must be associated
to specific roles or departments



Organizational models



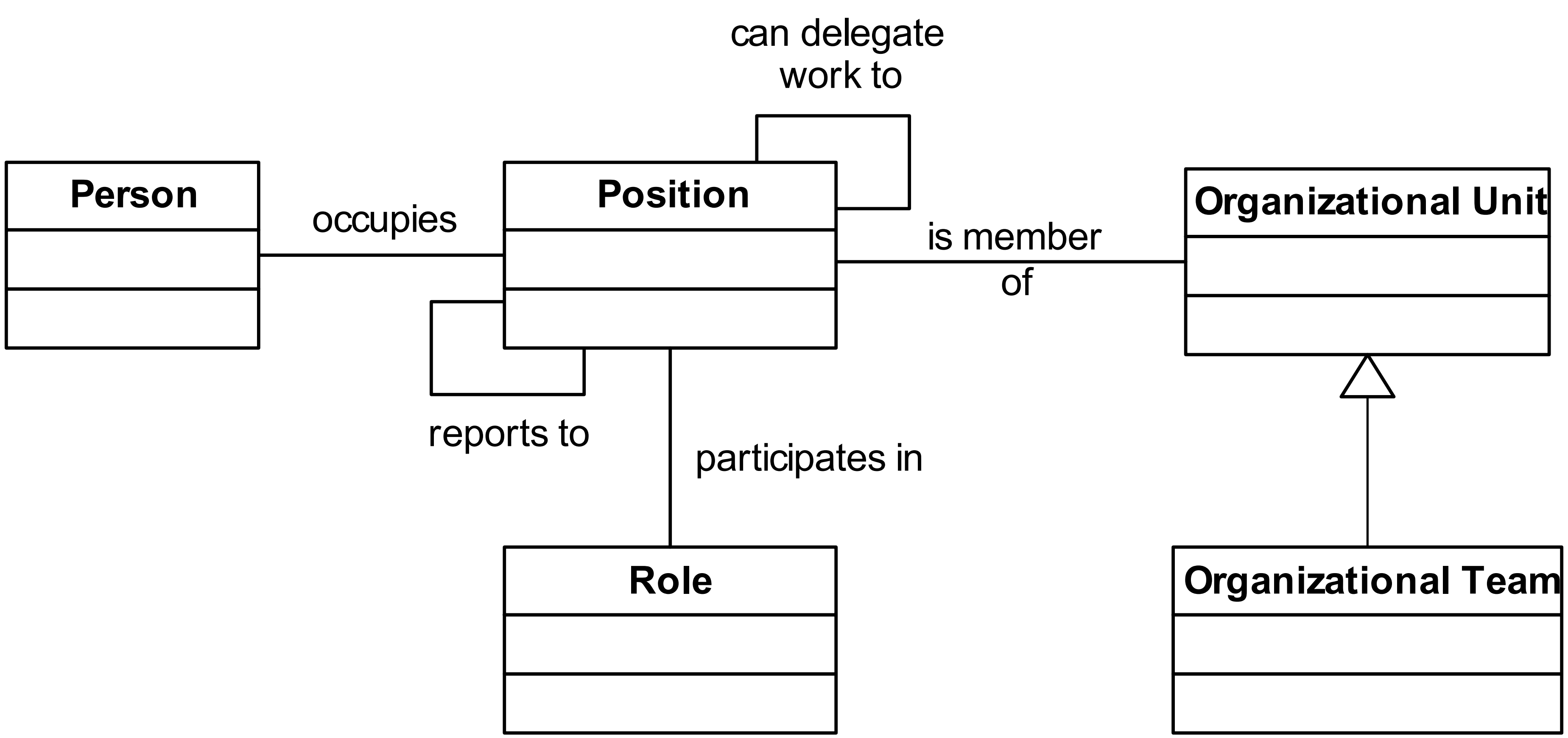


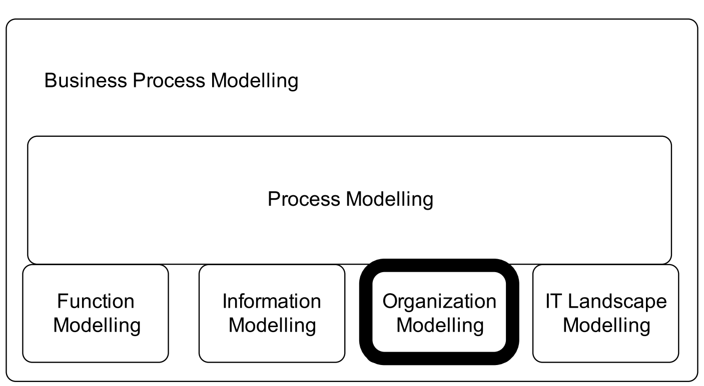
Roles

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

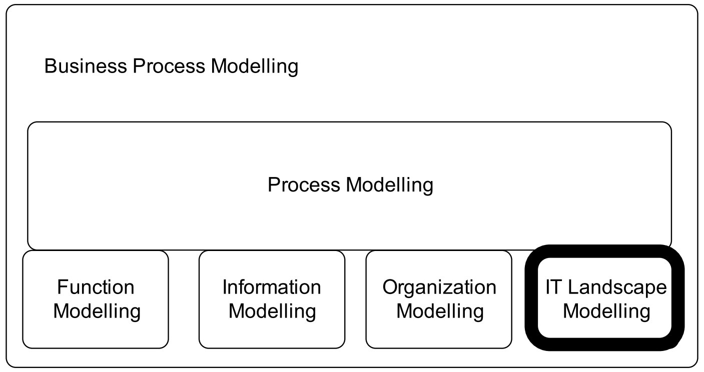




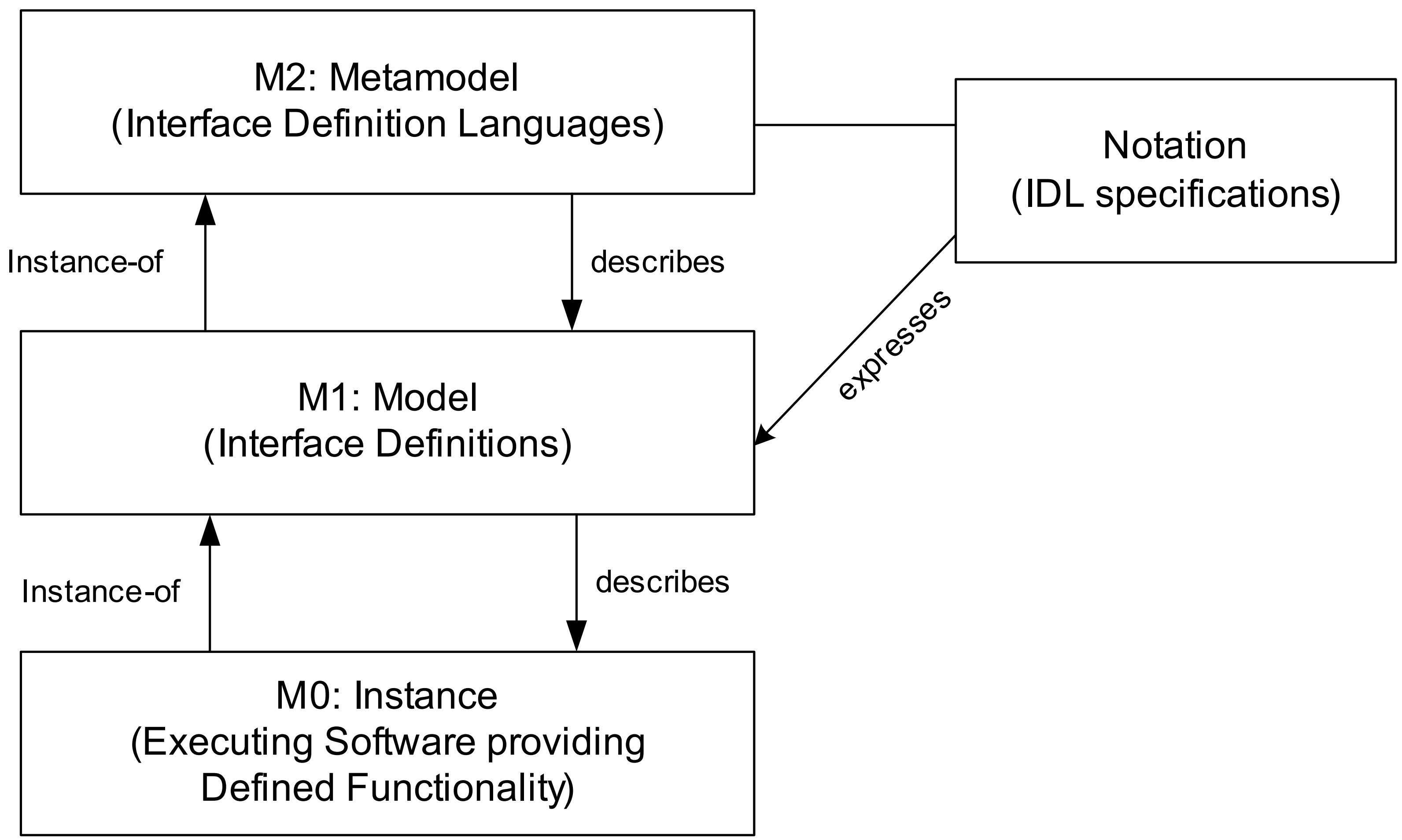
IT landscape

Many activities in a business process are supported by information systems

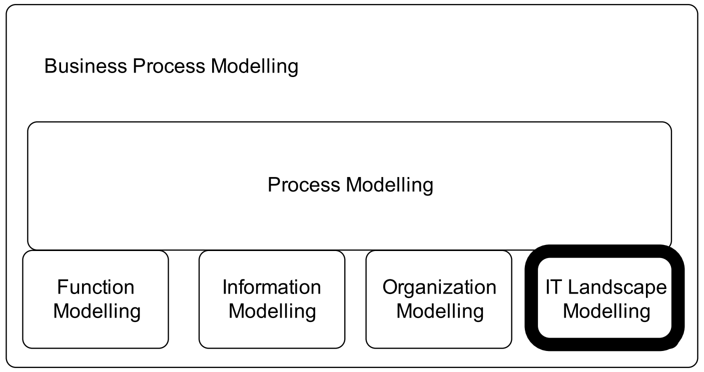
Information systems and programming interfaces needs to be represented because they provide functionalities



Interface Definition Languages



M. Weske: Business Process Management,

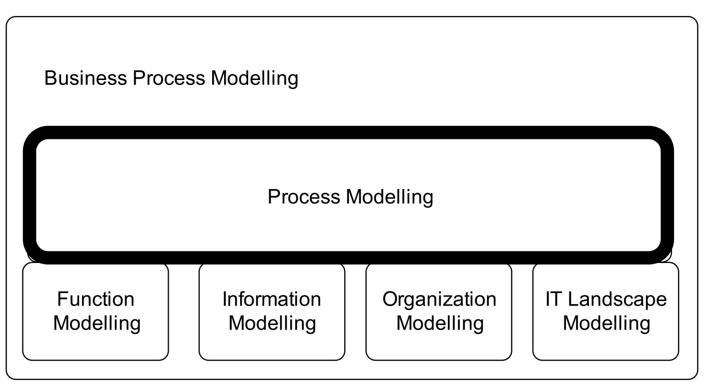


Process models

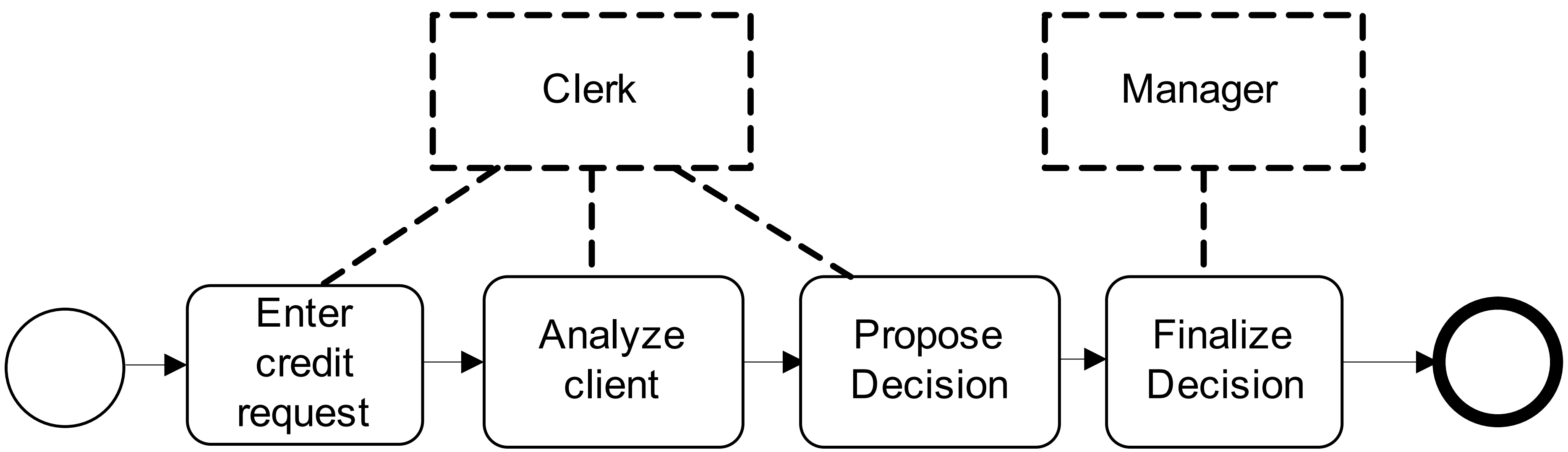
Define the glue between the subdomains

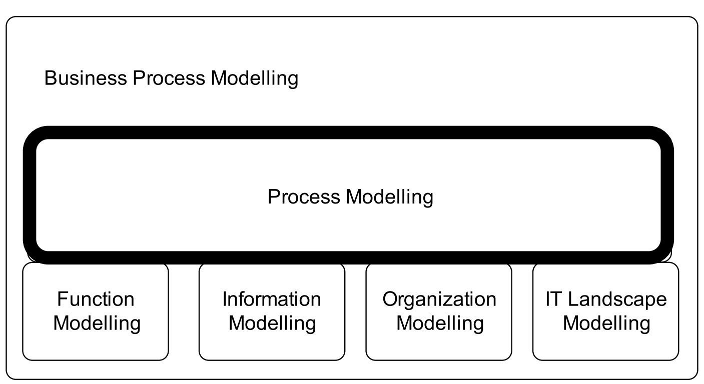
Relate functions and execution constraints

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



A process model with role information





A process instance with workers information