Business Processes Modelling

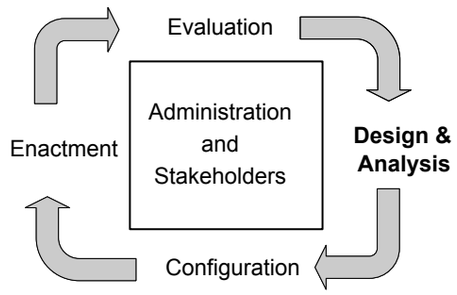
MPB (6 cfu, 295AA)

Roberto Bruni

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

05 - BP Lifecycle

Object



Overview the business process lifecycle

Ch. 1.2, 3 of Business Process Management: Concepts, Languages, Architectures

Lifecycle

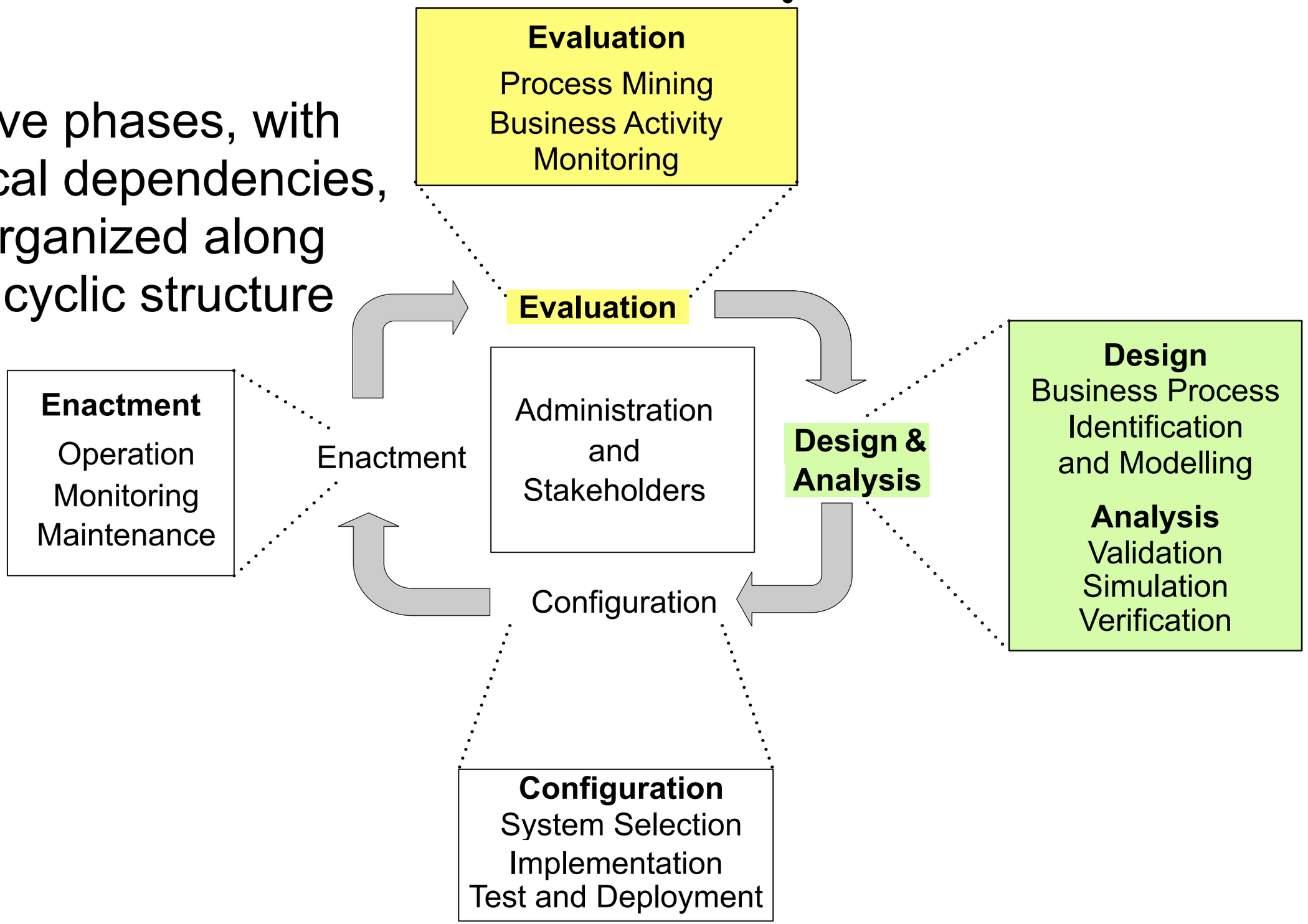
A lifecycle model is a conceptual description of the steps that are involved in building a product

The steps in which the model is broken are called **phases** (logically consistent, easier to understand)

The number of phases can vary from model to model (typically ranging from four to eigth)

BP lifecycle

Five phases, with logical dependencies, organized along a cyclic structure

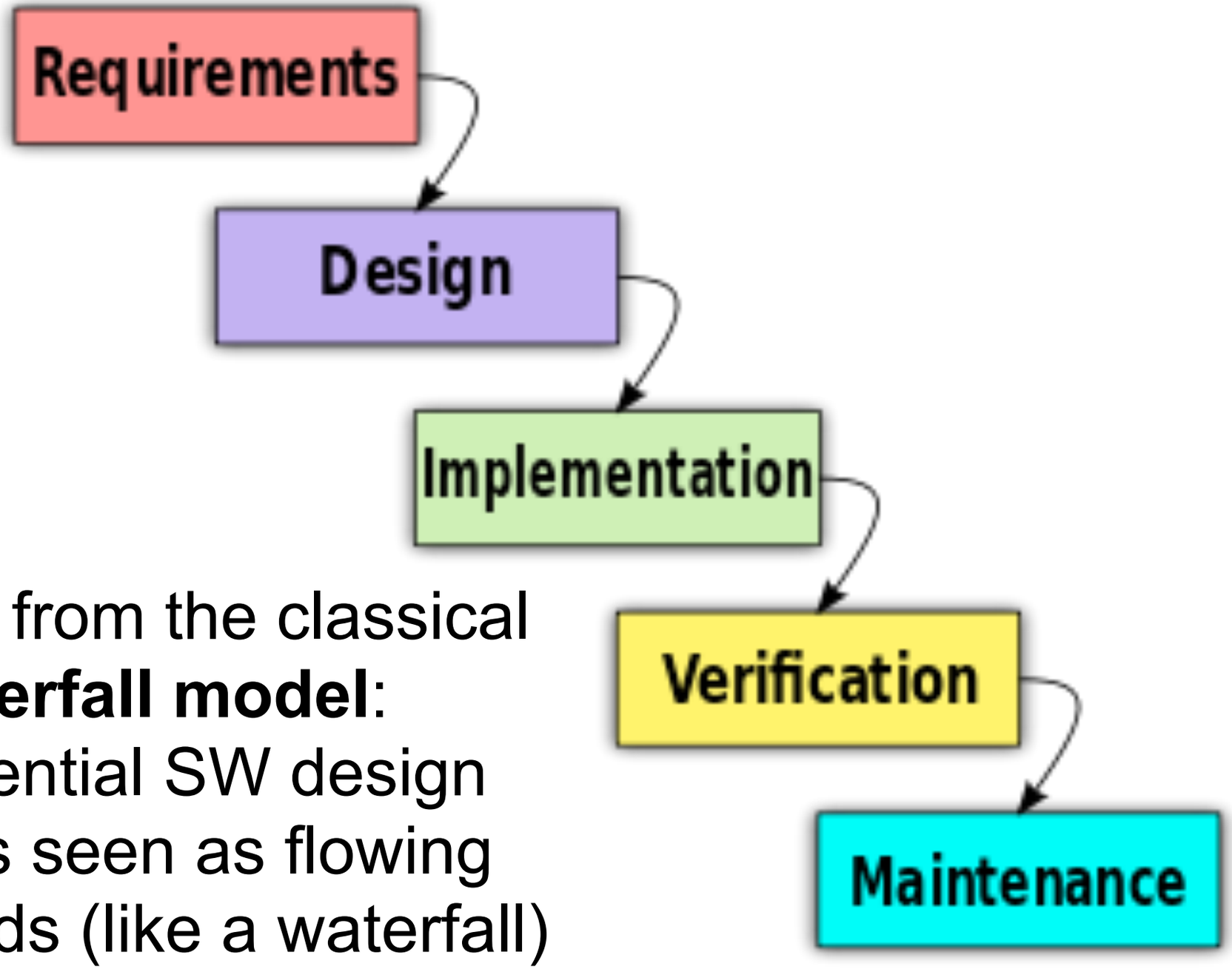


BP lifecycle

The logical dependencies between different phases **do not imply a strict temporal ordering** of their execution

Incremental and evolutionary approaches involving concurrent activities in multiple phases are frequently used

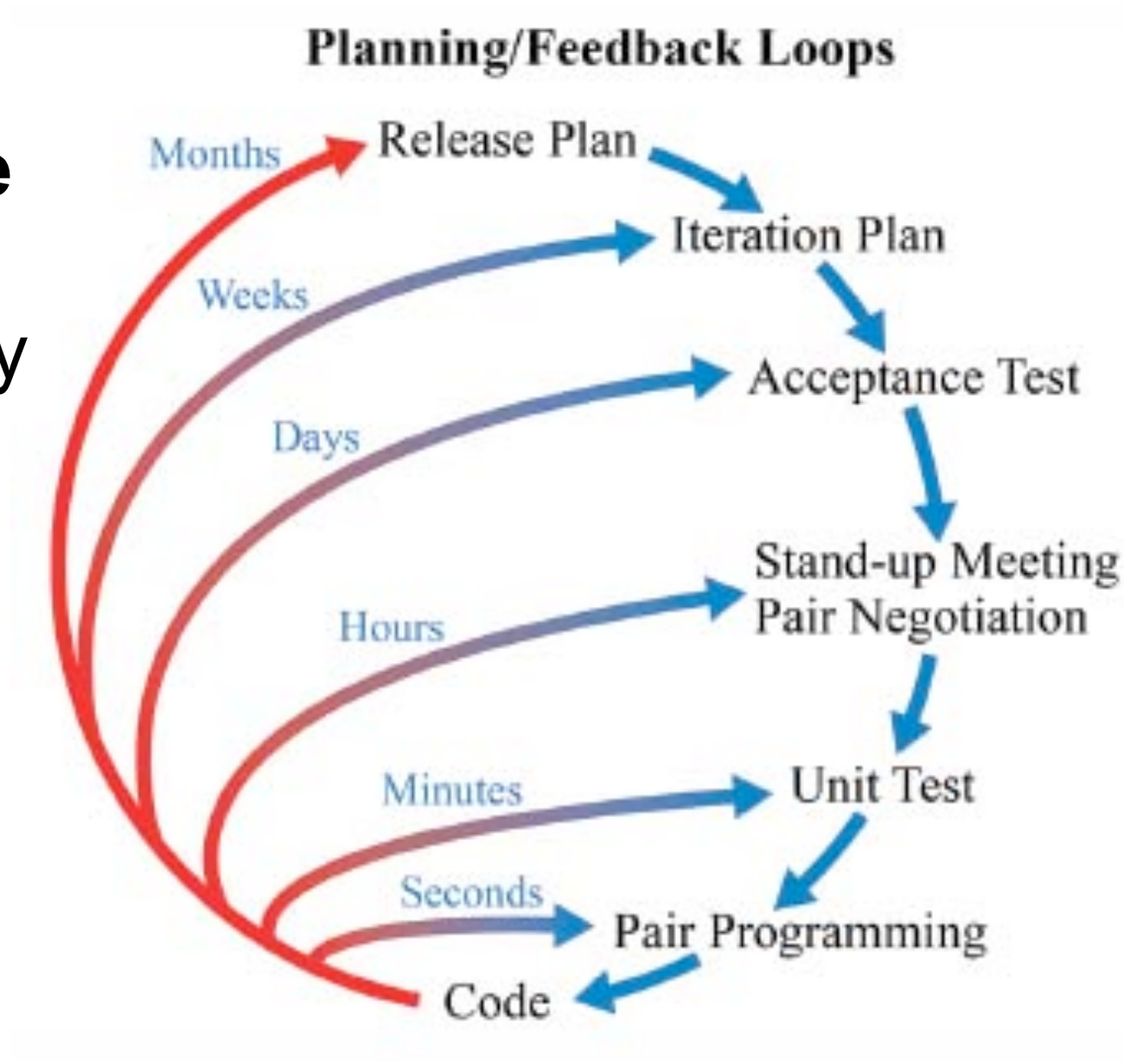
BP lifecycle vs waterfall



Different from the classical **waterfall model**:
a sequential SW design process seen as flowing downwards (like a waterfall) through various phases.

BP lifecycle vs XP

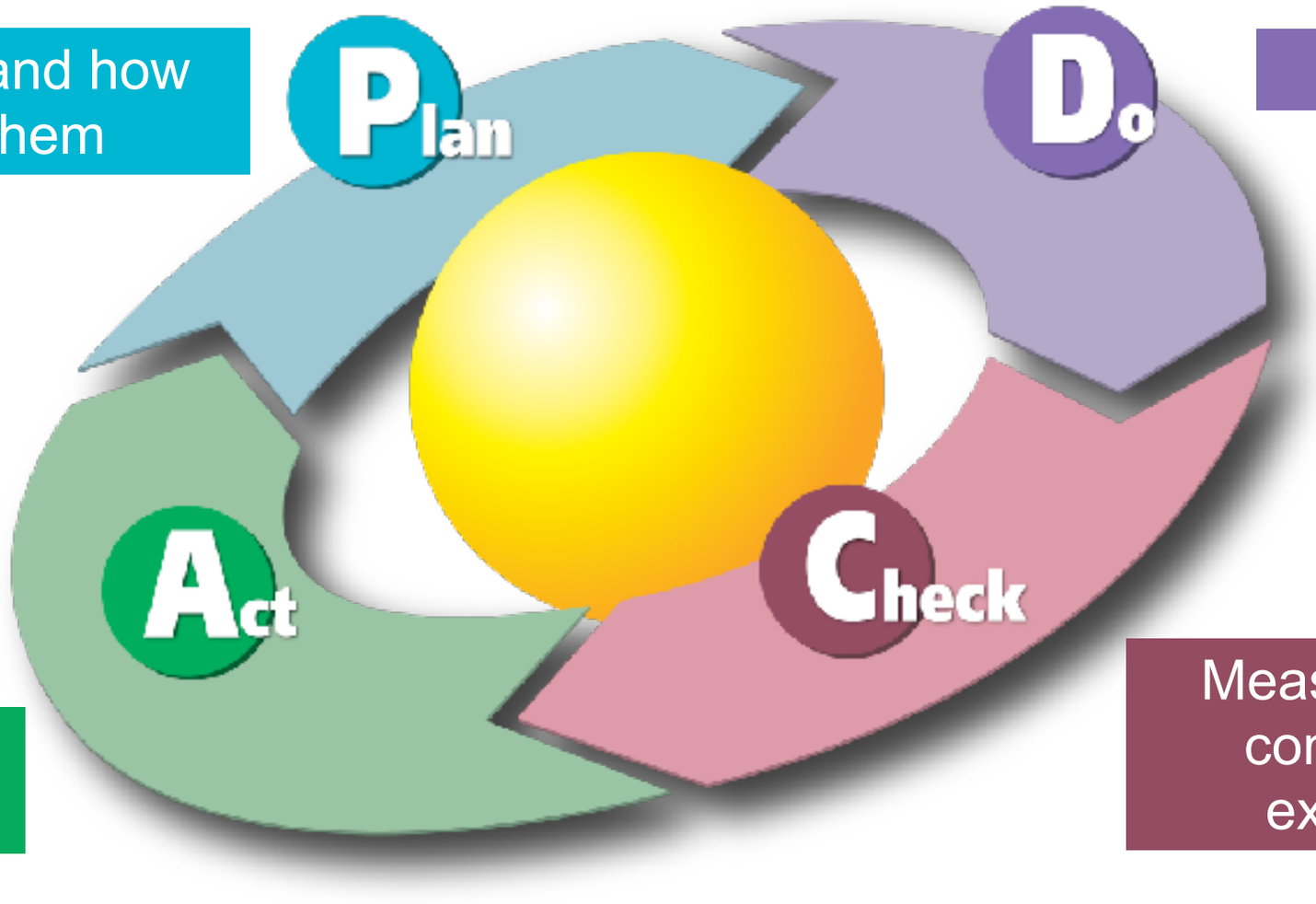
Better structured than **extreme programming** methodology: intended to improve productivity and responsiveness to changing requirements, advocates frequent releases, adding features when needed and a flat management structure



BP lifecycle vs PDCA

Similar to the **PDCA** scheme
(you may have heard of):
a management method for the control and
continuous improvement of products

Fix objectives and how to achieve them

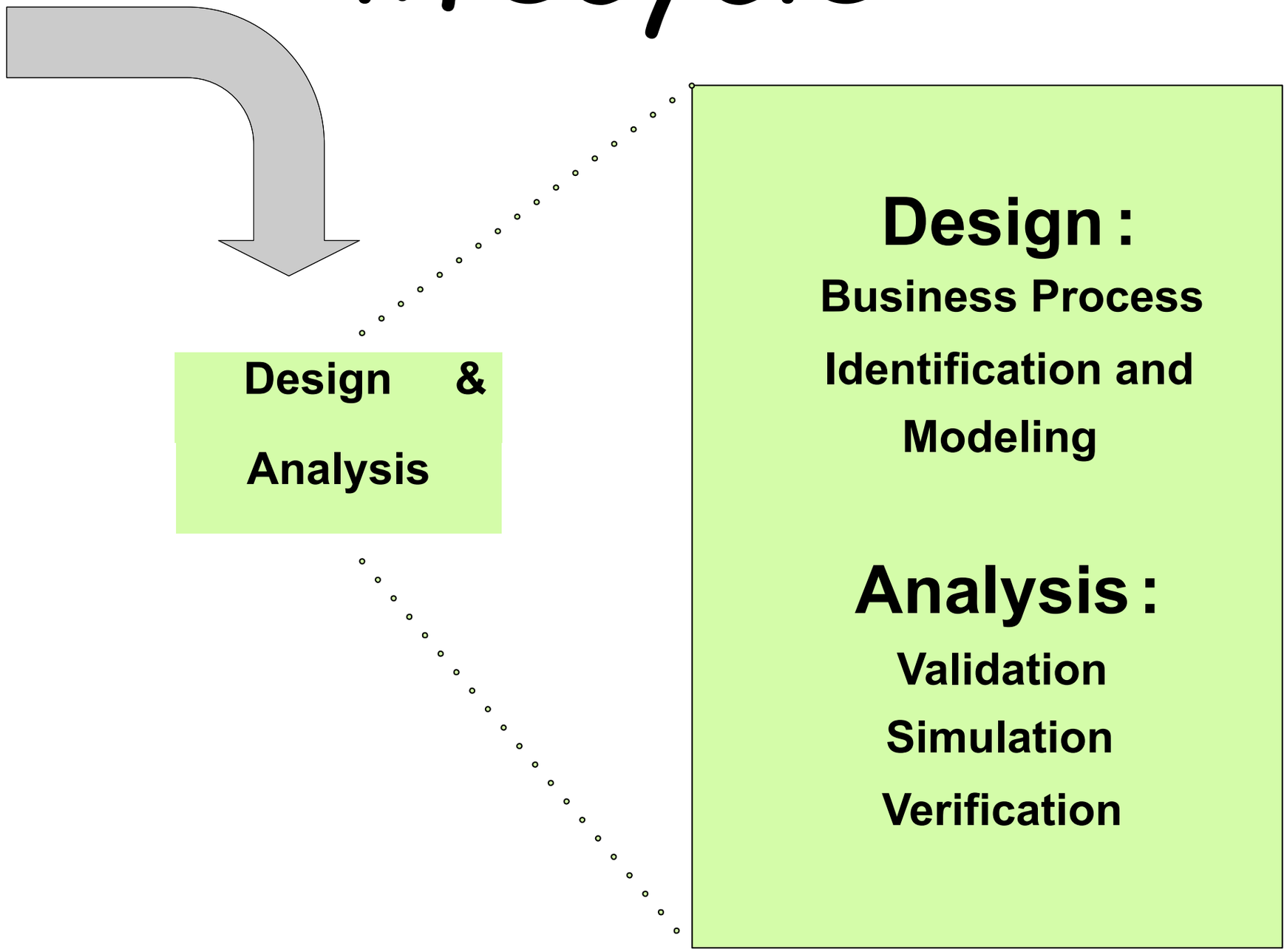


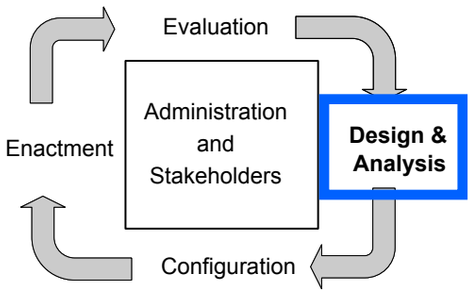
Execute plan

Measure results and compare them to expected goals

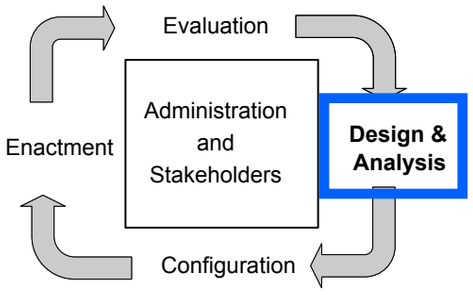
Detect issues and adjust goals

Business process lifecycle




Evaluation
Administration
and
Stakeholders
Design &
Analysis
Enactment
Configuration

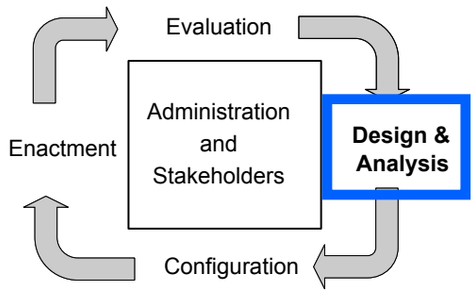
Design



Design: Identification

Require **surveys** on:
the business processes
their organizational environment
their technical environment

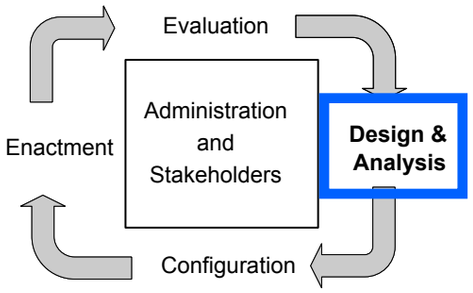
Based on these surveys, business processes are:
identified
reviewed
validated
represented (by business process models)



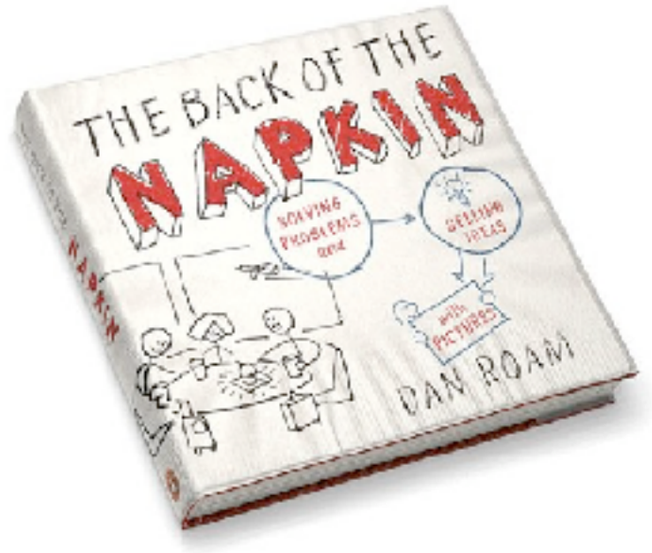
Design: Modeling

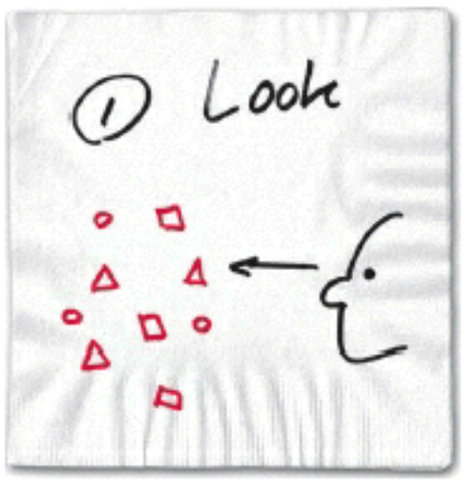
Core technical sub-phase:
from informal descriptions
to a particular business process modelling notation

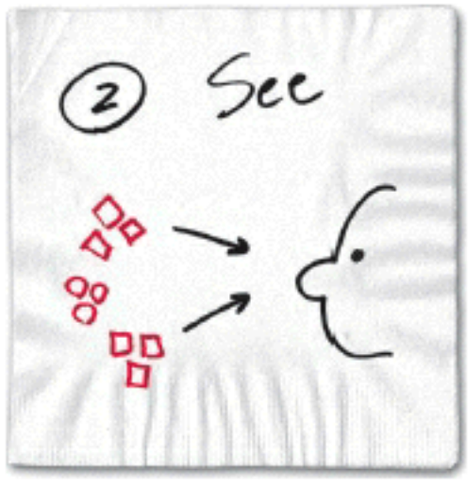
Explicit business process models expressed in a graphical notation facilitate communication about these processes so that different stakeholders can:
communicate efficiently
refine them
improve them

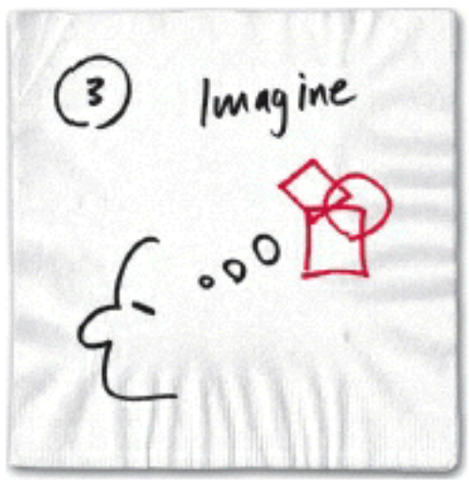


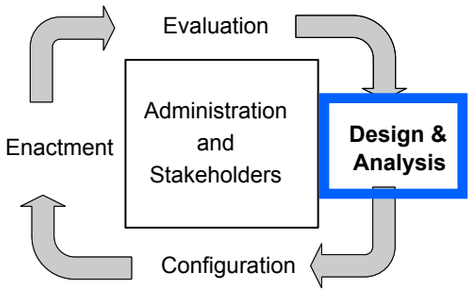
Look, see, imagine, show



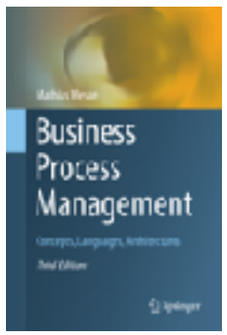






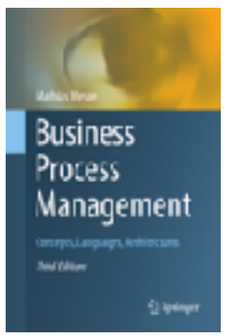


Business process model and instances



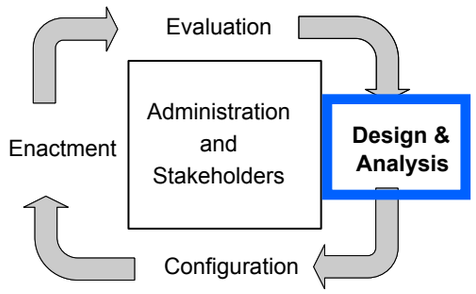
Definition: **business process model** consists of a set of activity models and execution constraints between them.

- Weske



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

- Weske



Model and instances

Enter credit request

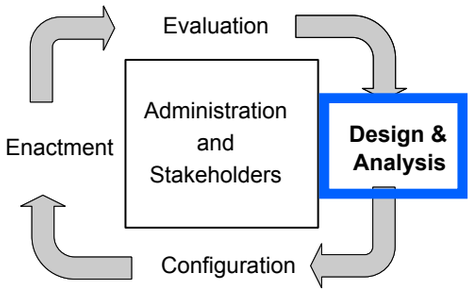
Enter c.r. (r017, Miller , 10000)

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

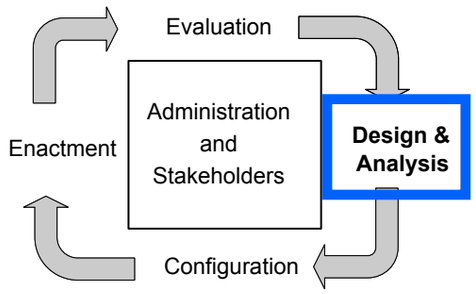
Each business process model acts as a blueprint
for a set of business process instances
(related to cases)



Abuse of notation

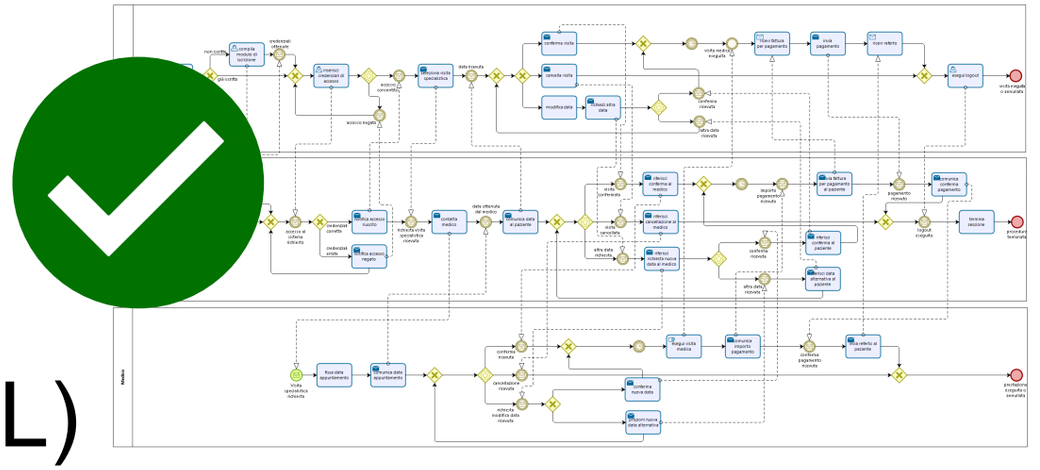
If no confusion is possible,
the term **activity** is used to refer
to activity models as well as activity instances

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

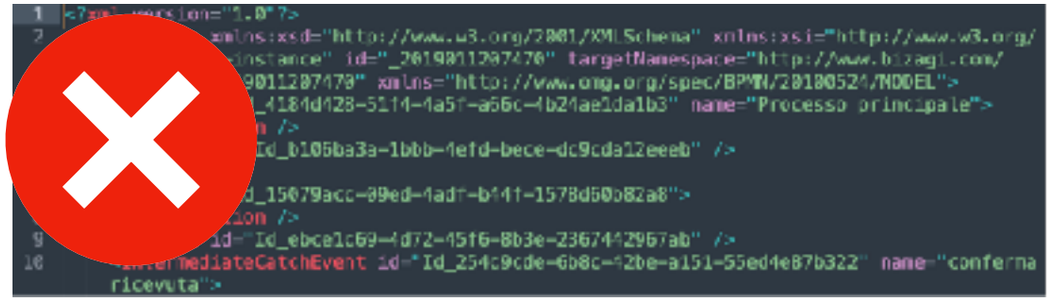


Representing processes

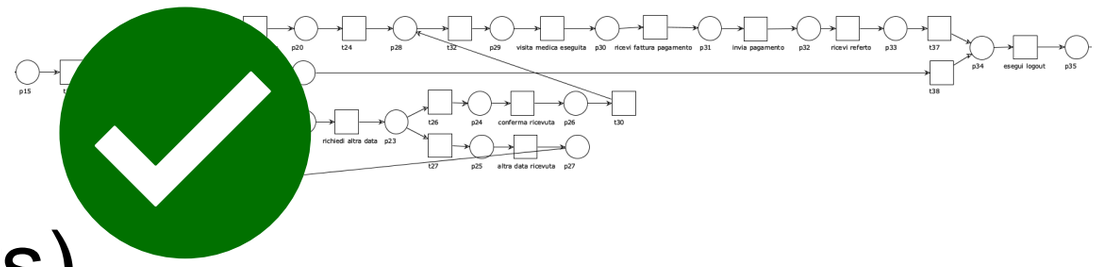
Visual representations:
diagrams and charts
understandable by humans
(informal, intuitive, BPMN, EPC, BPEL)

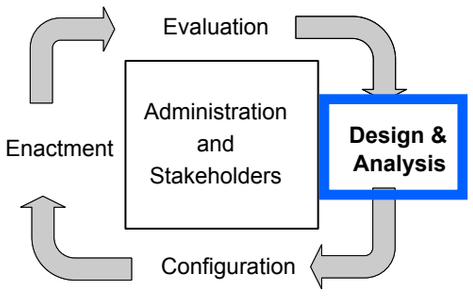


Languages:
unambiguous machine syntax
(process dialects, XML schemes)



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



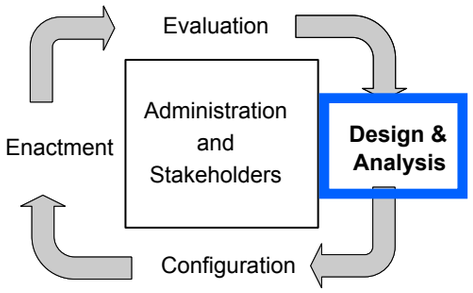


Do you know XML?

eXtensible Markup Language:
file format for storing and transmitting data

XML tags represent the data structure and contain metadata

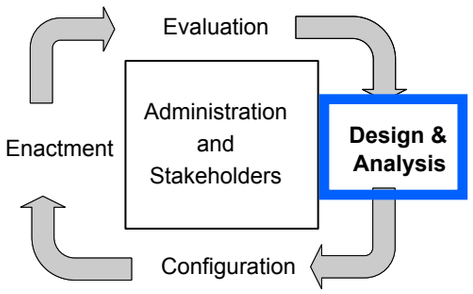
```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Bob</to>
  <from>Alice</from>
  <heading>Reminder</heading>
  <body>Don't forget to buy oranges!</body>
</note>
```



Models

A model is a simplified representation of reality

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



Abstraction

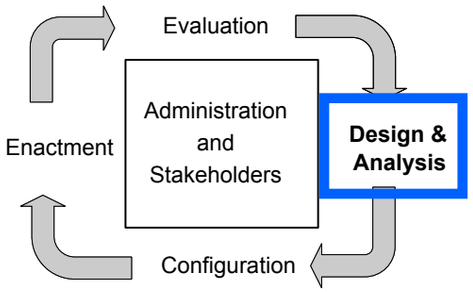
To derive general rules and concepts
from specific examples of some phenomenon,
by selecting only the aspects which are relevant
for a particular purpose

A way to cope with complexity

Horizontal: separation at different modeling levels

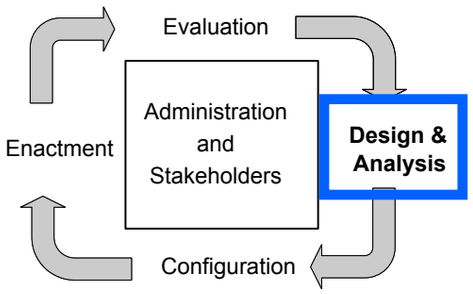
Aggregation: separation at different granularity levels

Vertical: separation at different subdomains



Modelling:

Horizontal Abstraction



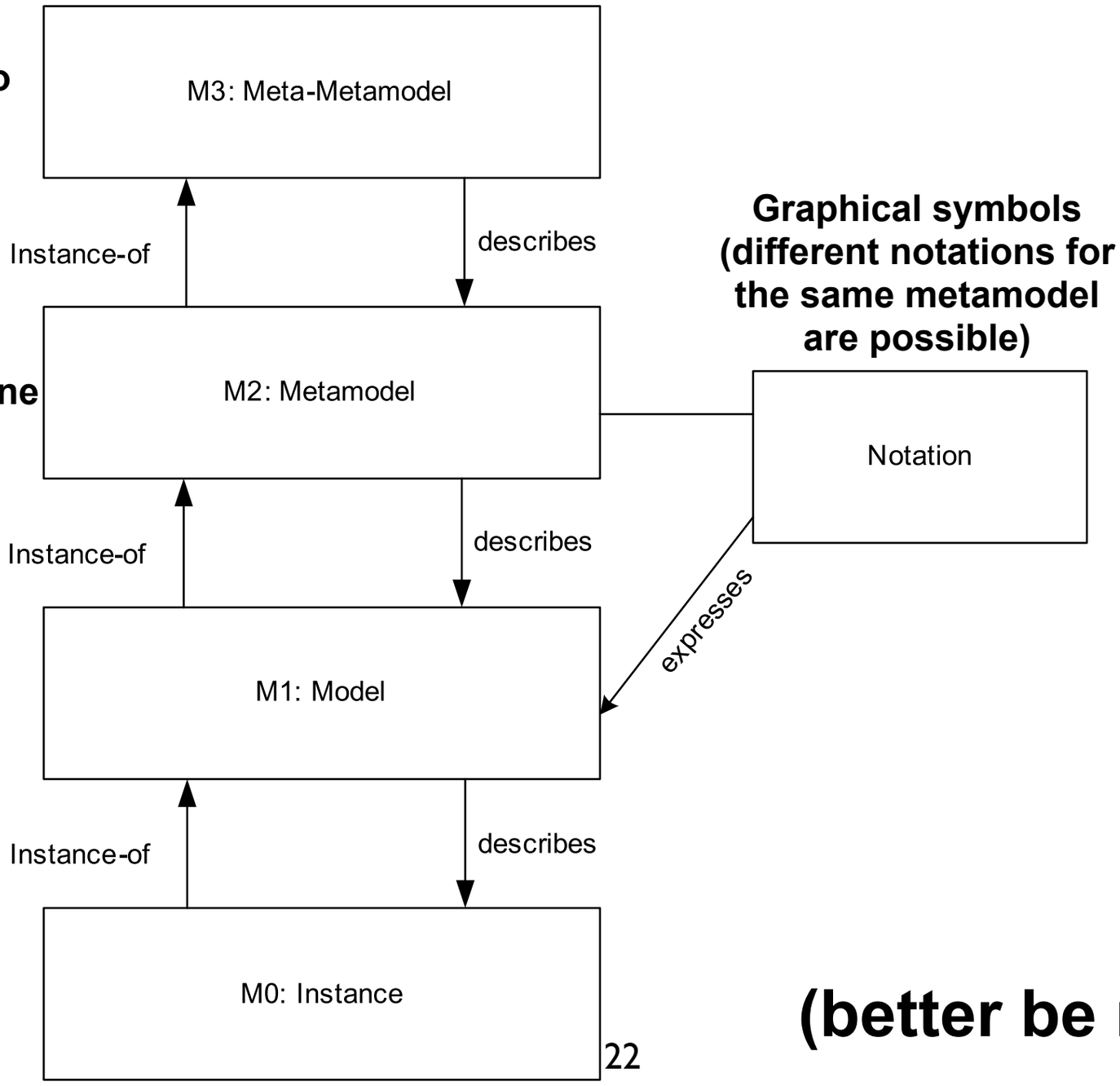
Horizontal abstraction (modeling levels)

Abstract entities to define concepts

Concepts that discipline model definition

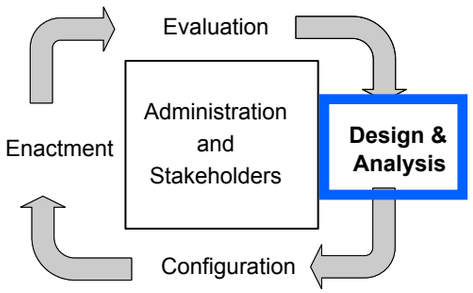
Classes of similar instances

Concrete entities

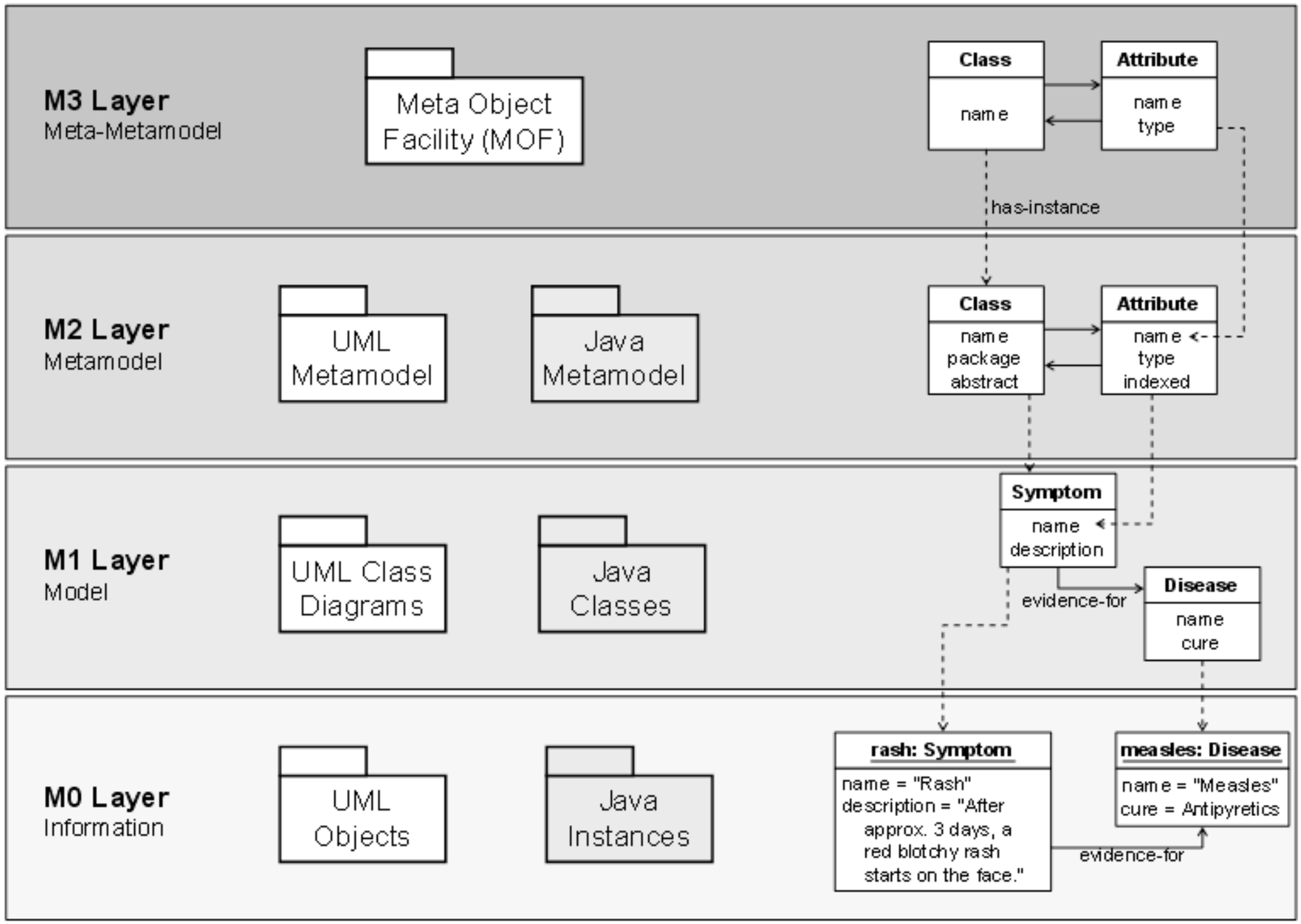


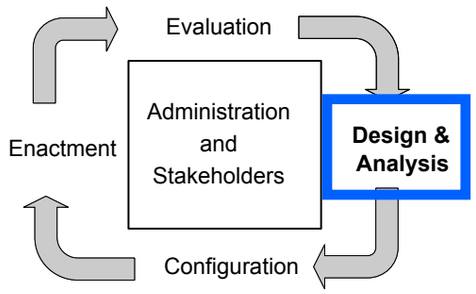
Graphical symbols (different notations for the same metamodel are possible)

(better be read bottom-up)

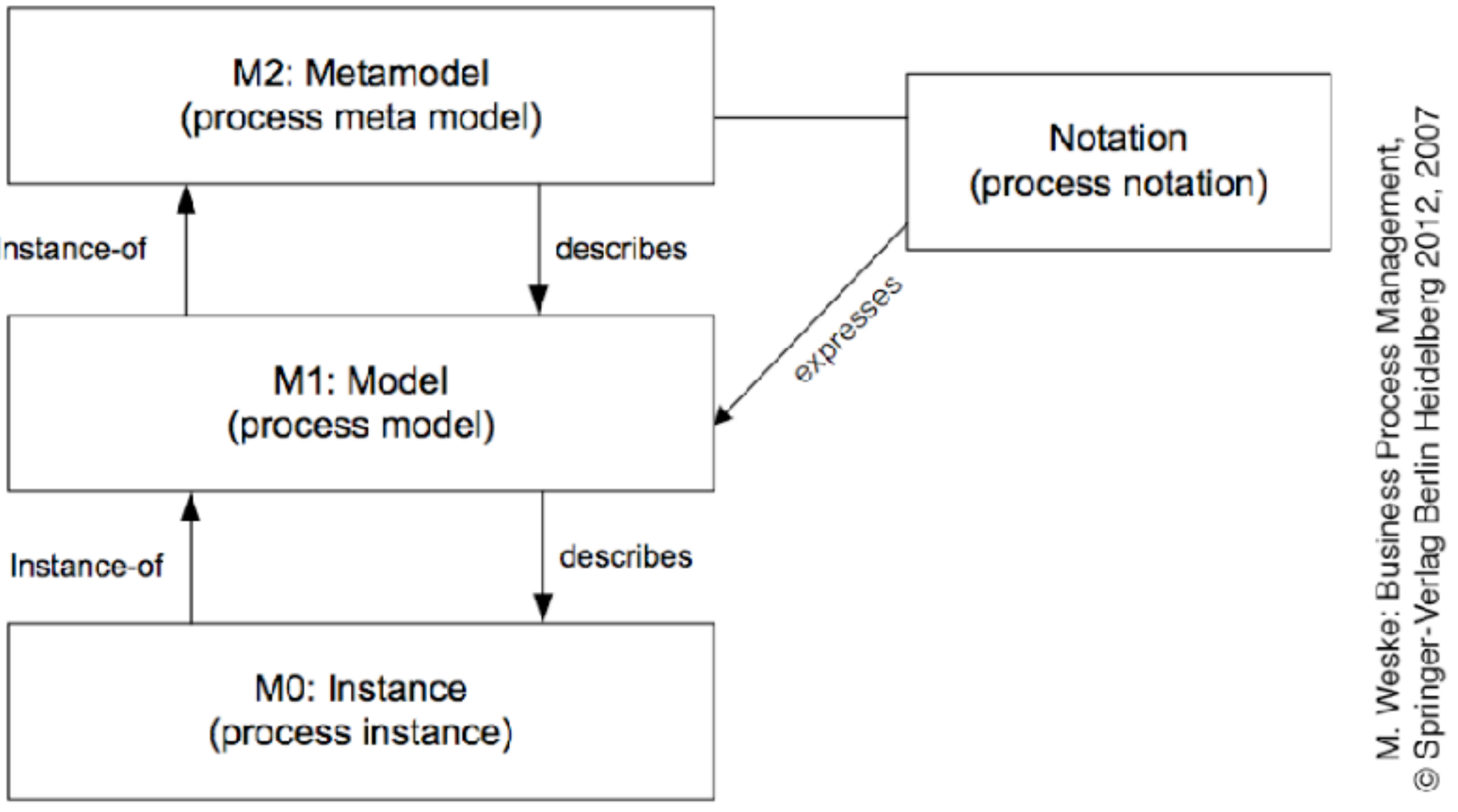


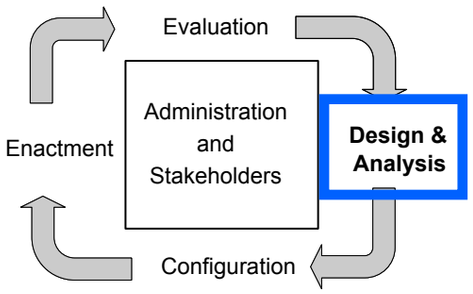
An example: MOF metamodel (OMG)





Process models and process instances

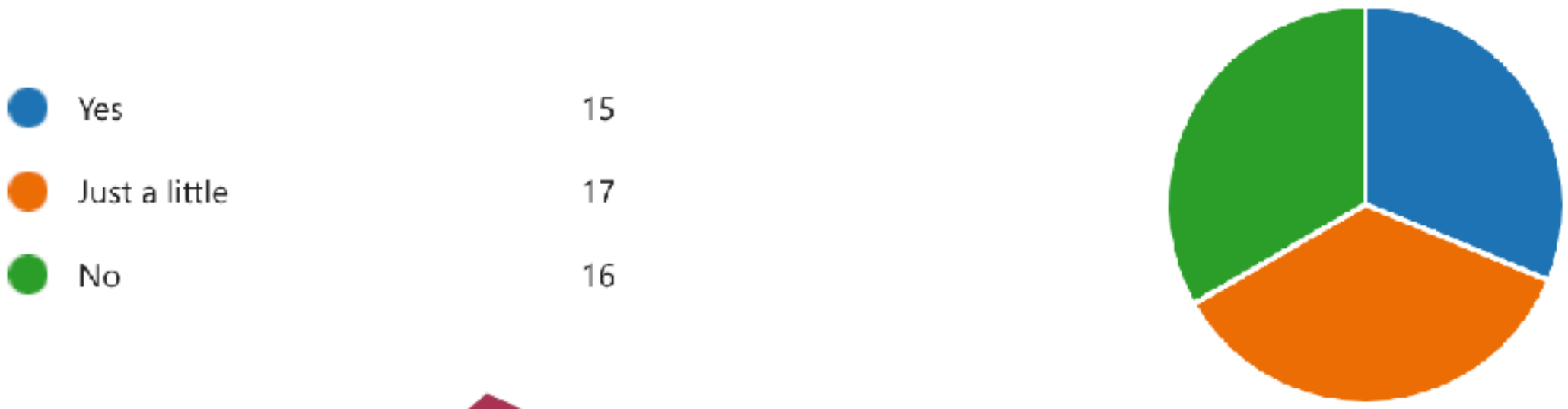




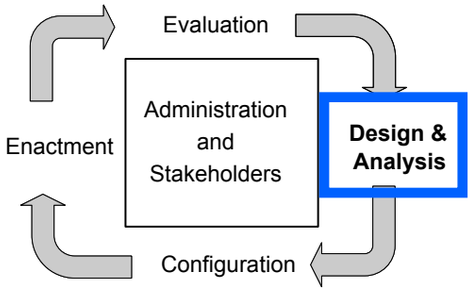
Do you know UML?

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

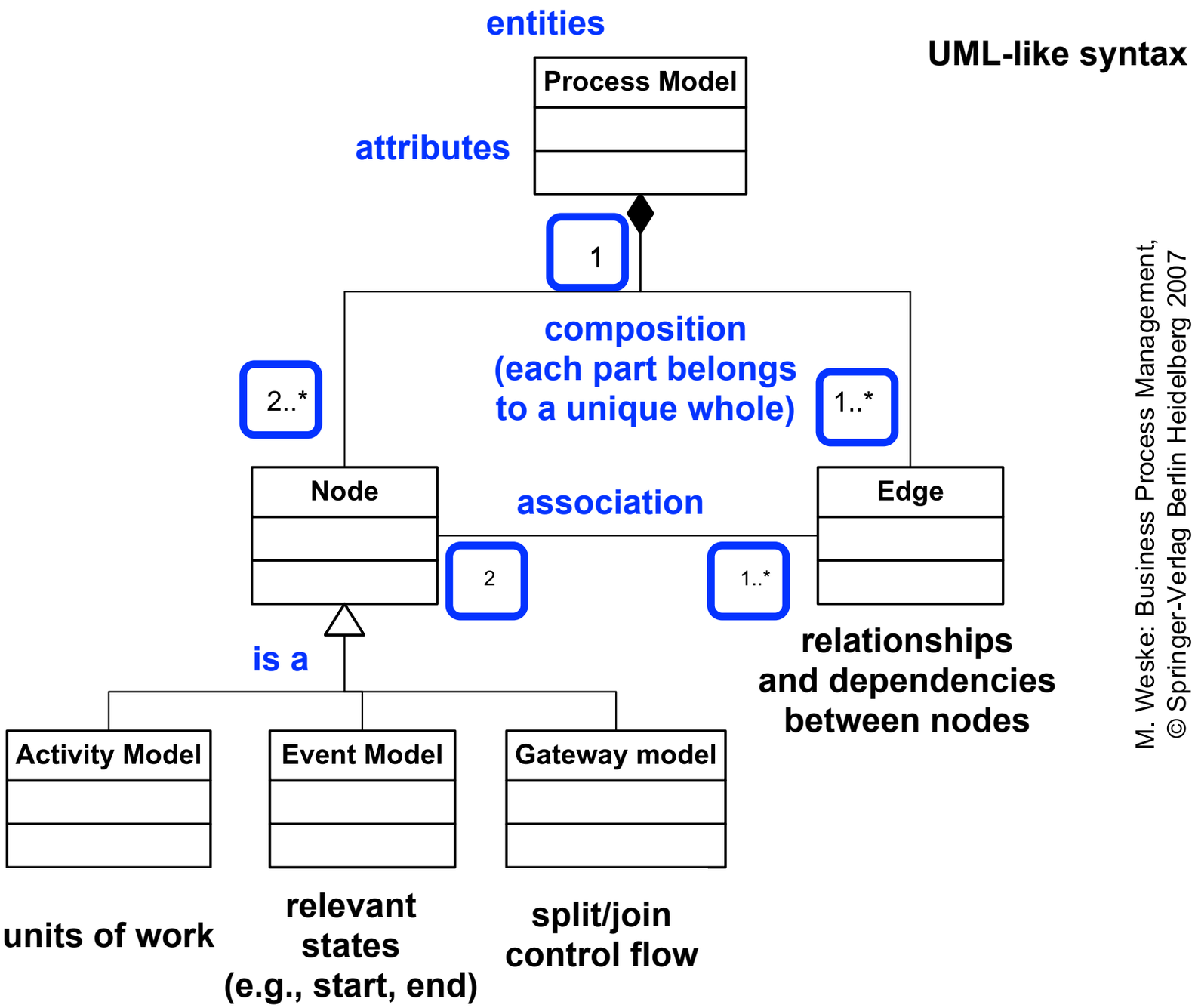
Altri dettagli

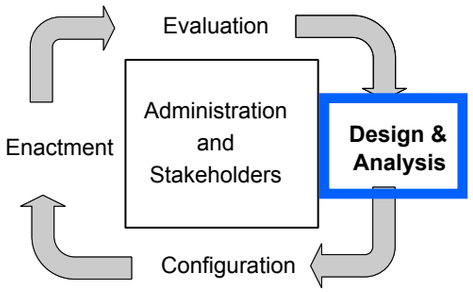


It is a general-purpose *visual modeling language* that is intended to provide a standard way to visualize the design of a system

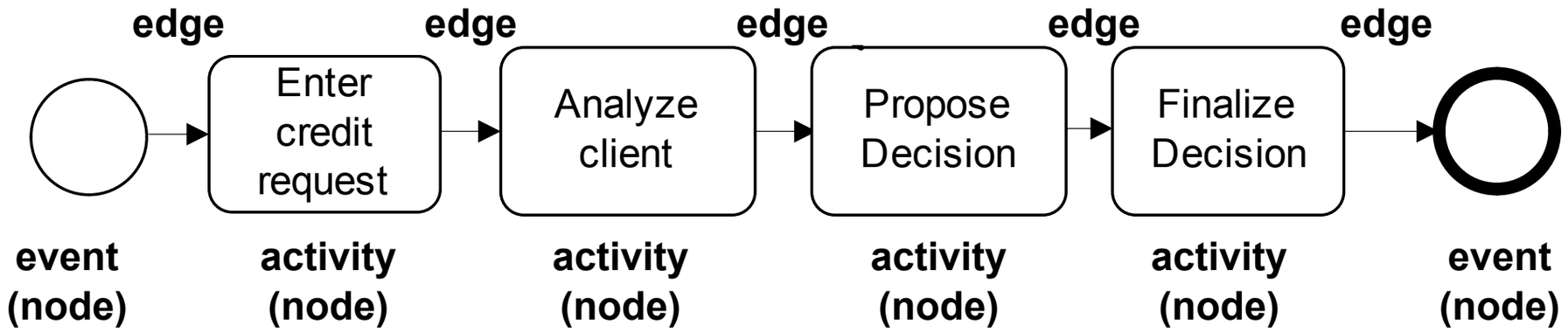


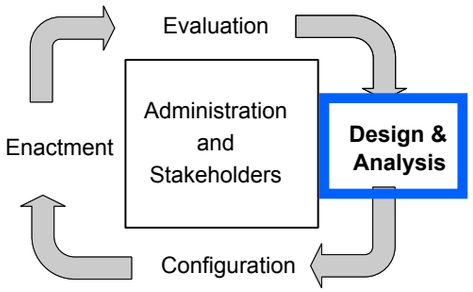
A process metamodel (level M2)



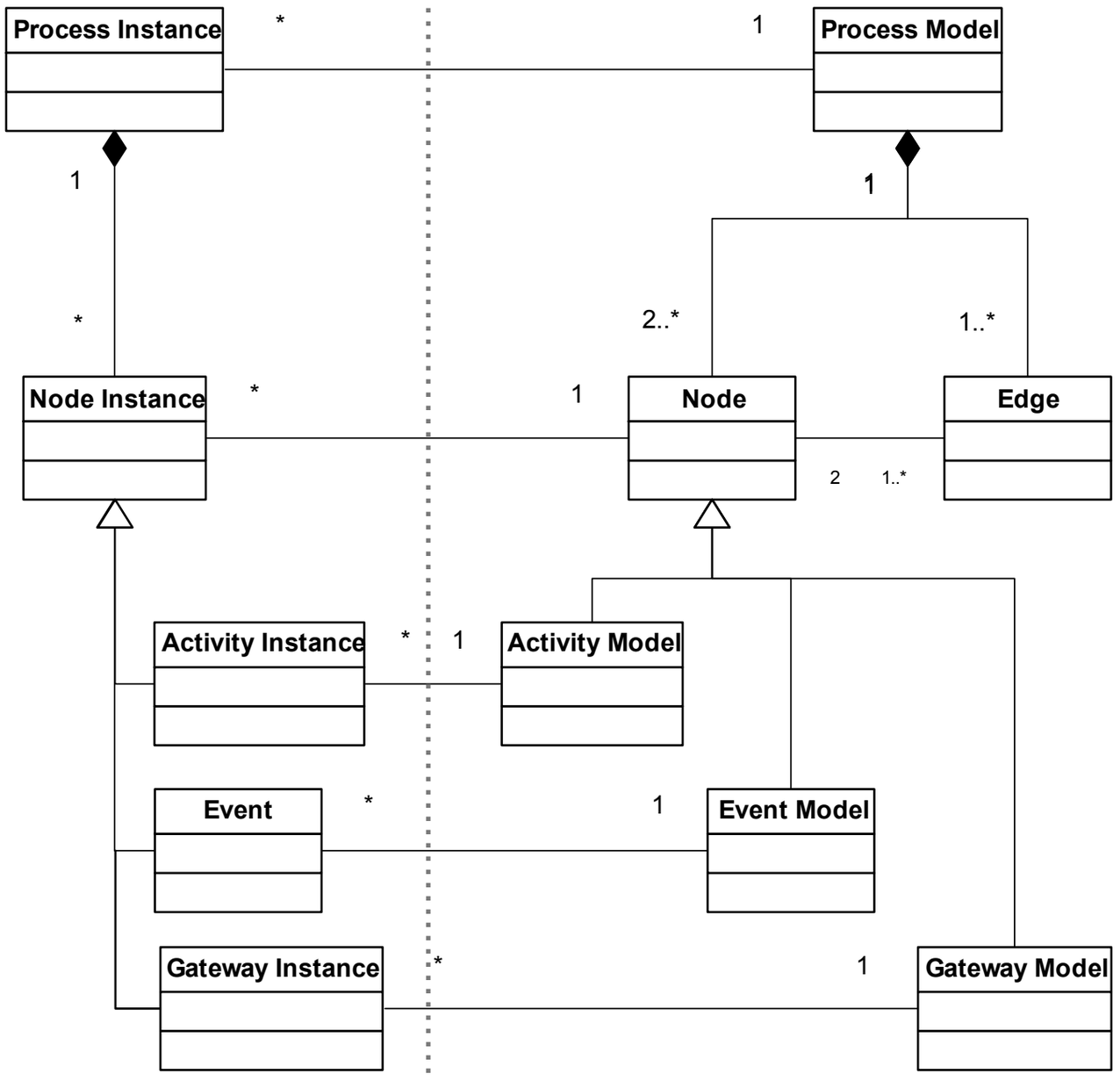


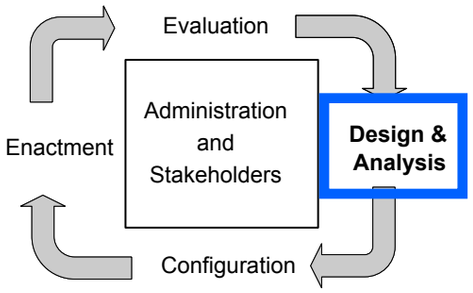
A process model



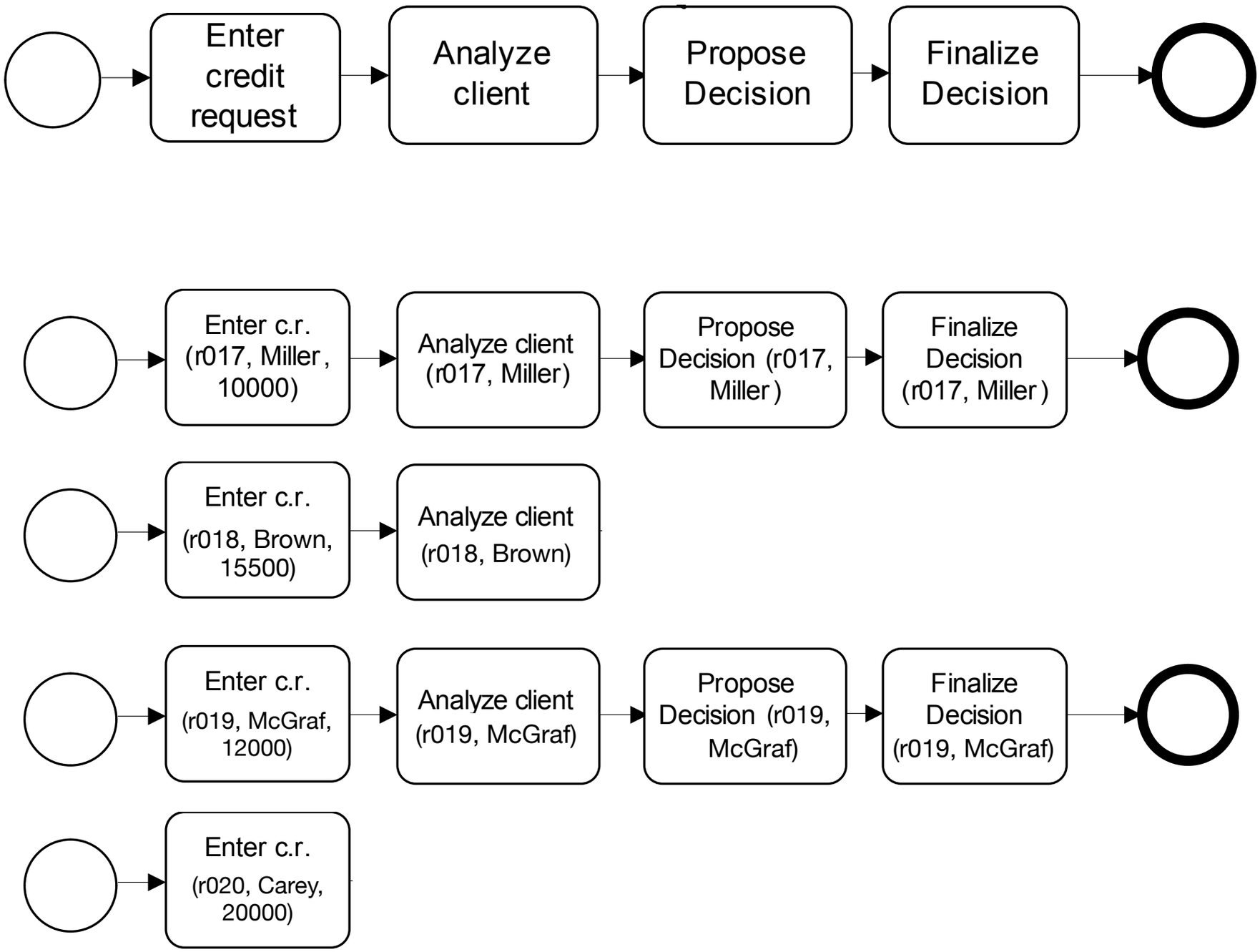


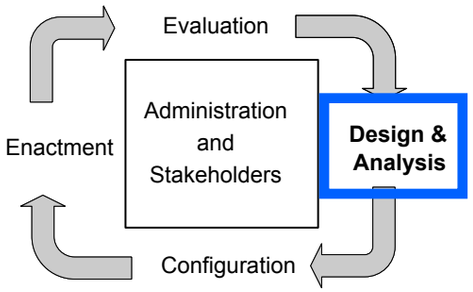
Process models and process instances





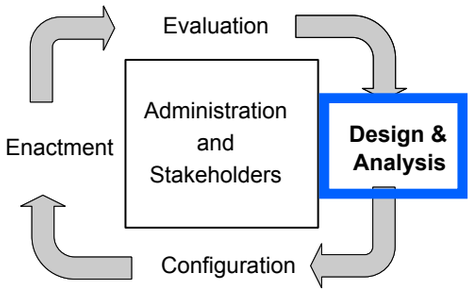
Some process instances





Modelling:

Aggregation Abstraction

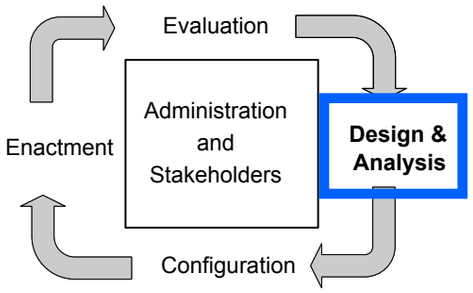


Aggregation abstraction

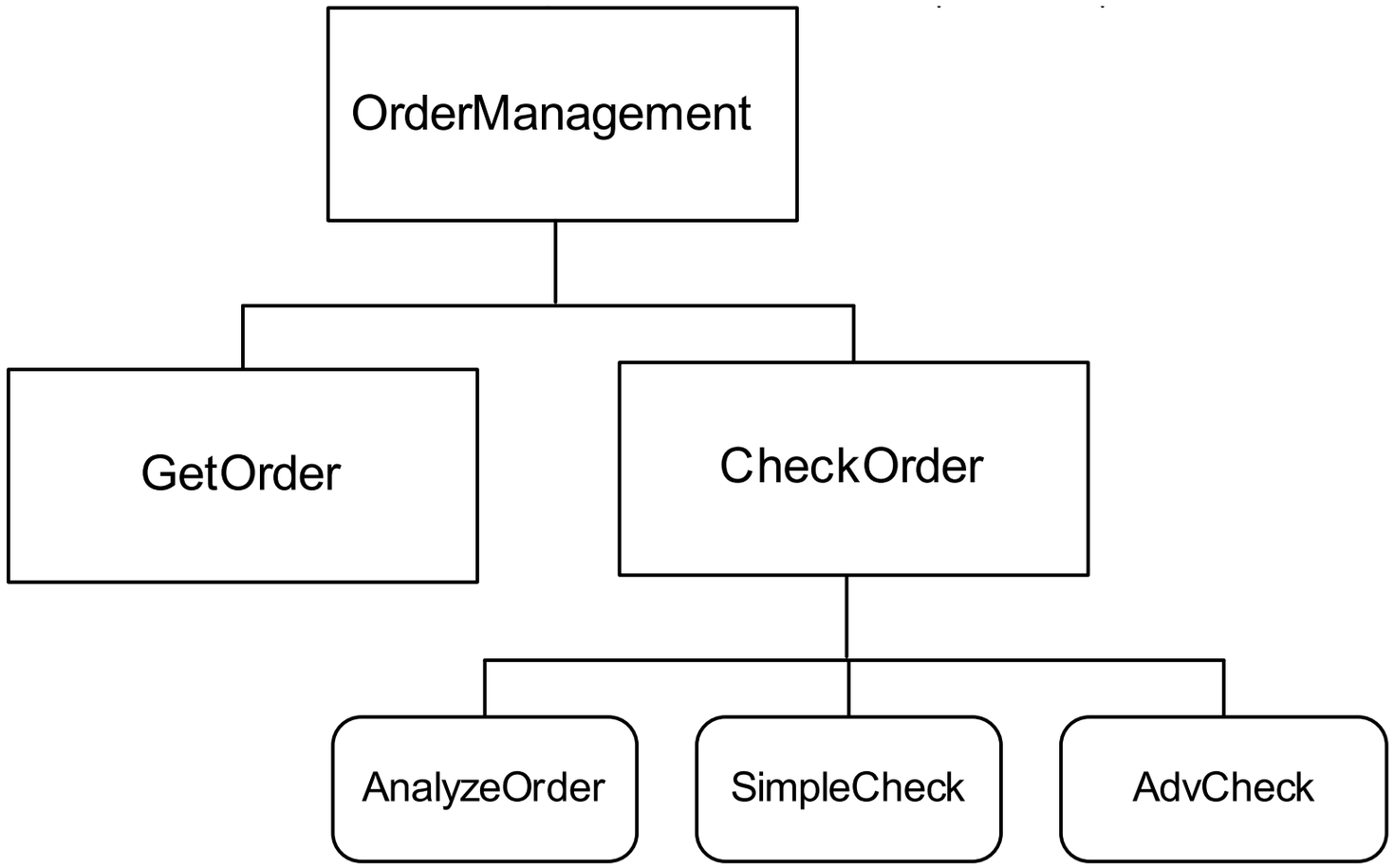
Multiple elements of a lower level of granularity can be grouped and represented by a single artefact at the higher level of granularity

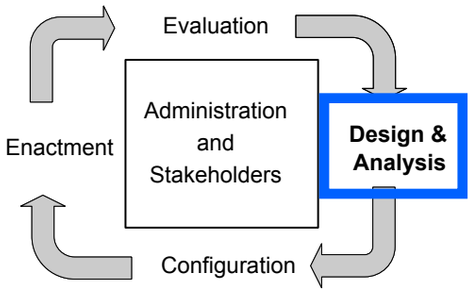
Different from horizontal abstraction, where all entities lie at the same level of granularity

Related to functional decomposition



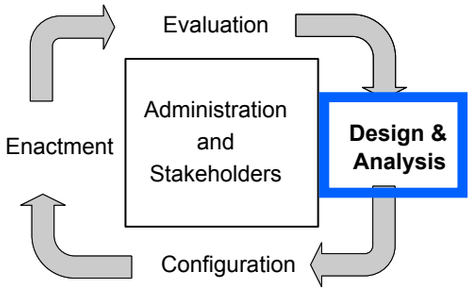
A sample aggregation





Modelling:

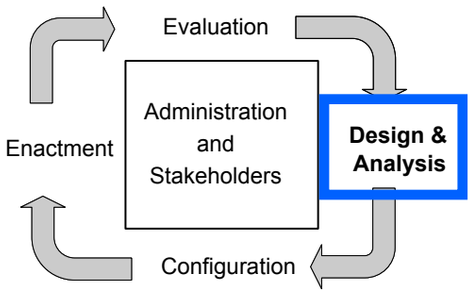
Vertical Abstraction



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*)

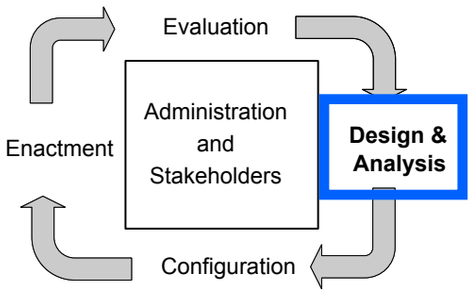
Edsger Wybe Dijkstra was one of the most influential members of computing science's founding generation. Among the domains in which his scientific contributions are fundamental are

- 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.

Look for EWD447:
On the role of scientific thought

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

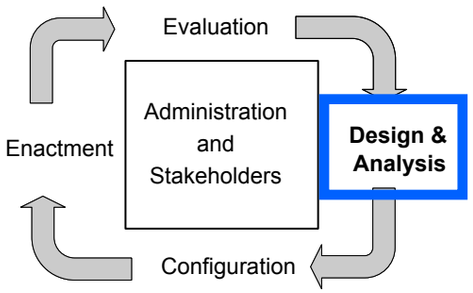


(EWD447)

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*.*

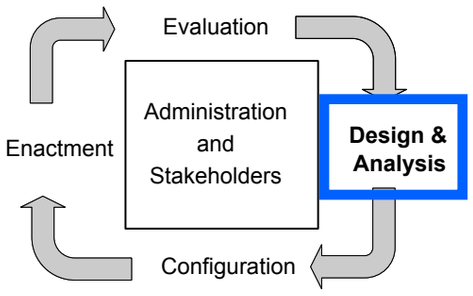


(EWD447)

*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.***

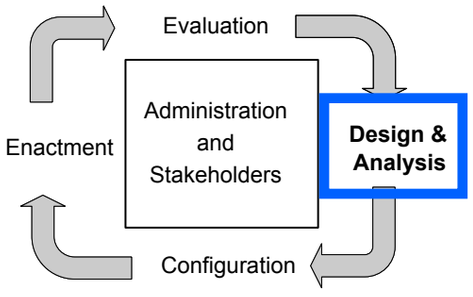


SoC: an example

HyperText Markup Language (HTML):
organization of webpage content

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

JavaScript (JS):
user interactions



Do you know HTML?

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

Altri dettagli

Yes	18
Just a little	15
No	15

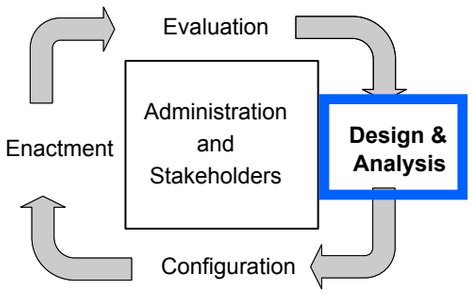
```html
<!DOCTYPE html>
<html>
<style>
body {
  background-color: lightblue;
}

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

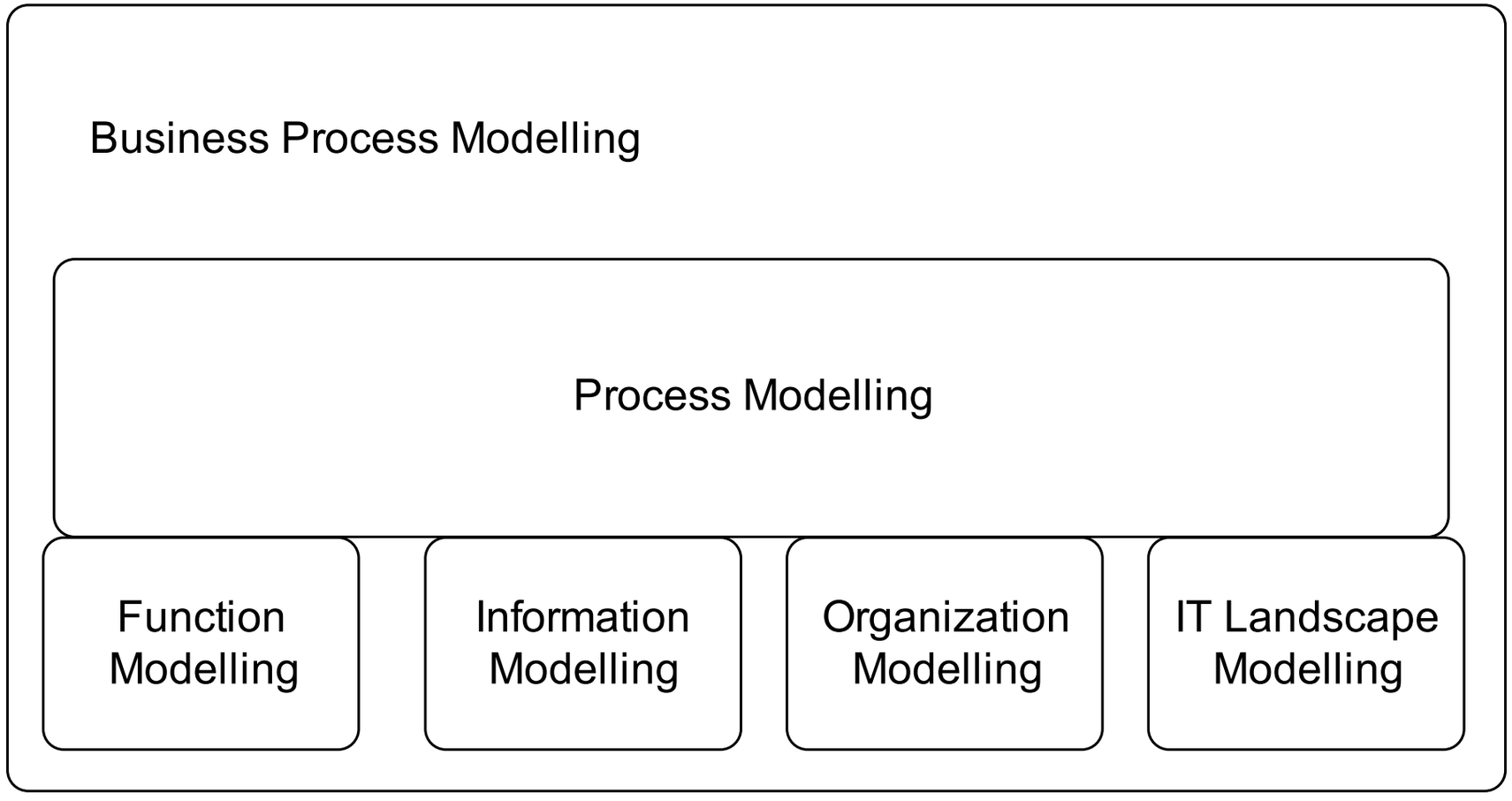

```html
<!DOCTYPE html>
<html>
<style>
body {
  background-color: lightyellow;
}

h1 {
  color: darkred;
  text-align: center;
}

p {
  font-family: courier;
  font-size: 15px;
}
</style>
```

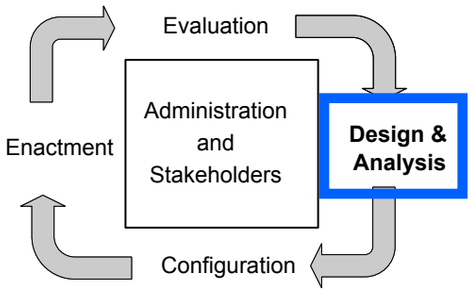
CSS



Vertical abstraction (domain separation)

BPM includes multiple modelling domains, integrated by Process Modelling



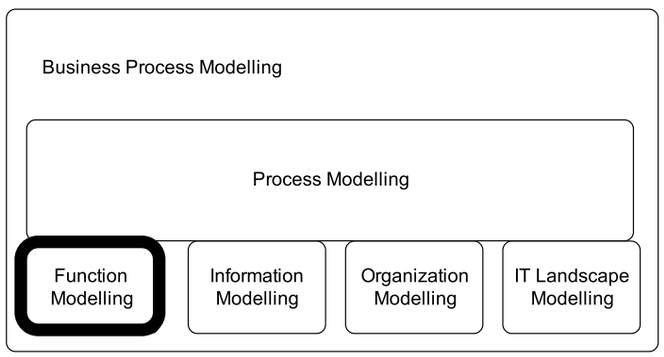


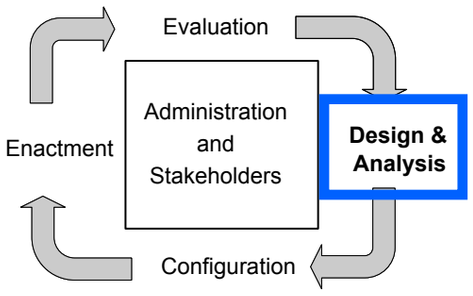
Function models

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)

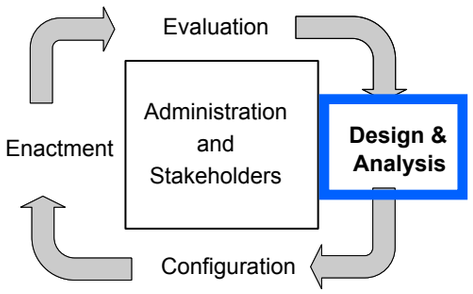




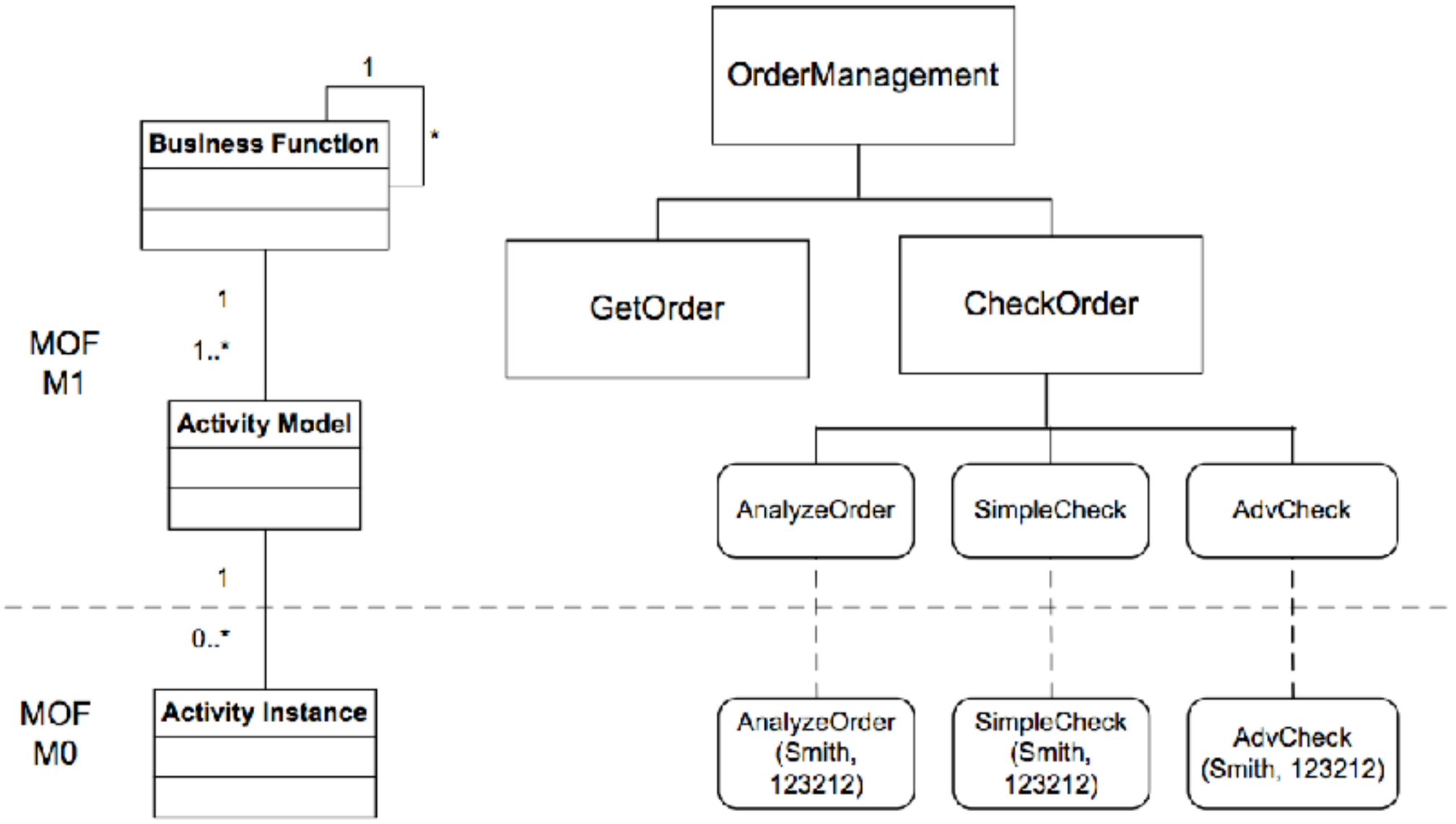
High-level business functions

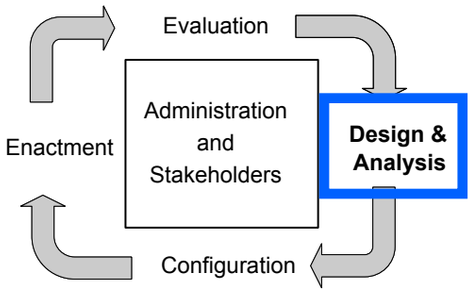
The value chain of a company has a rich internal structure, consisting of a set of coarse-grained business functions
(e.g. Order management, Human resources)

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

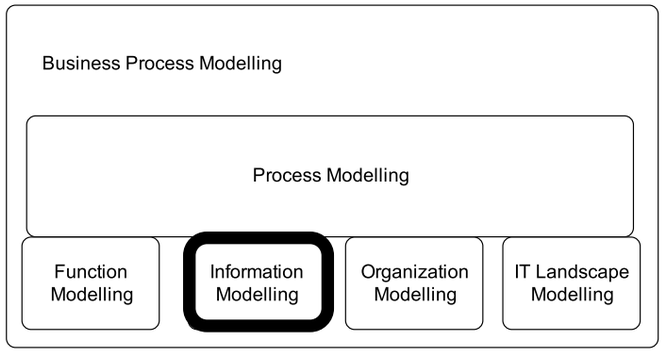


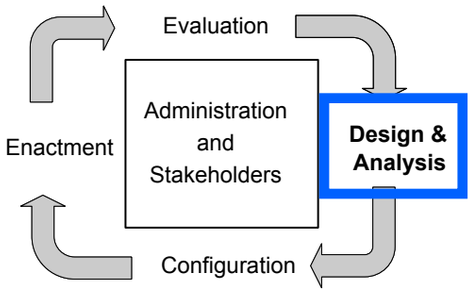


Information models

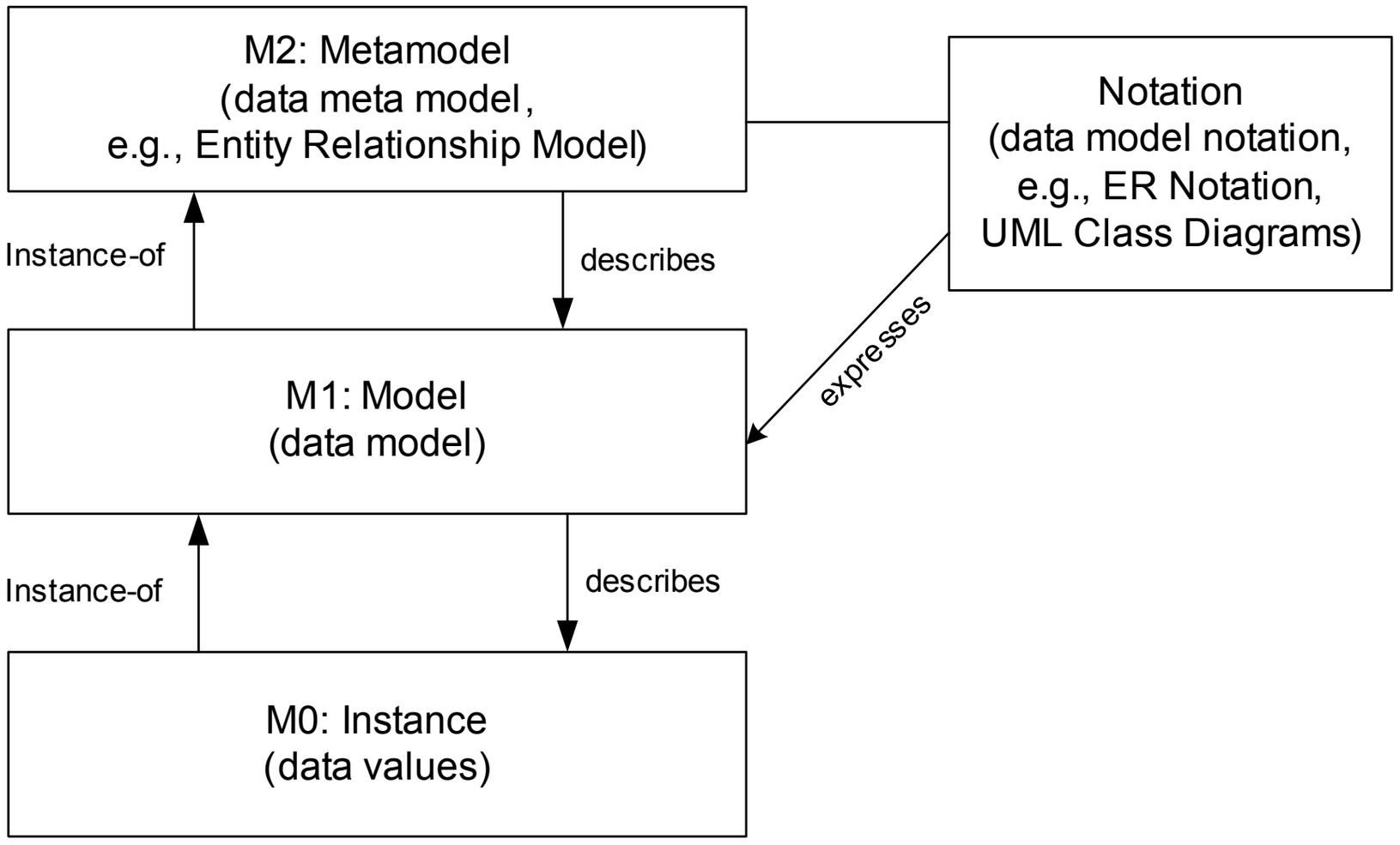
Data representation is crucial:
all decisions made during a business process depends on data values

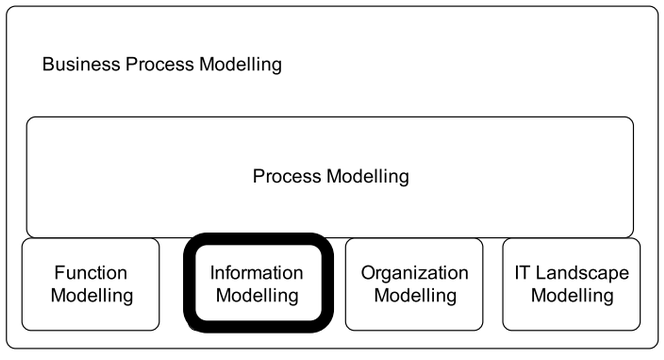
Data dependencies between activities are also important
(ensure data-availability, reduce waiting time)

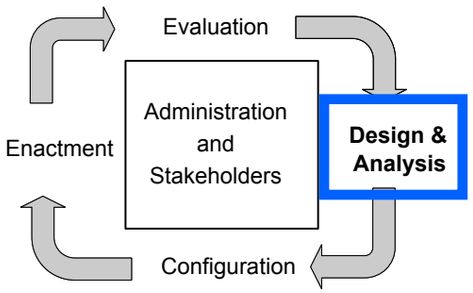




Data models





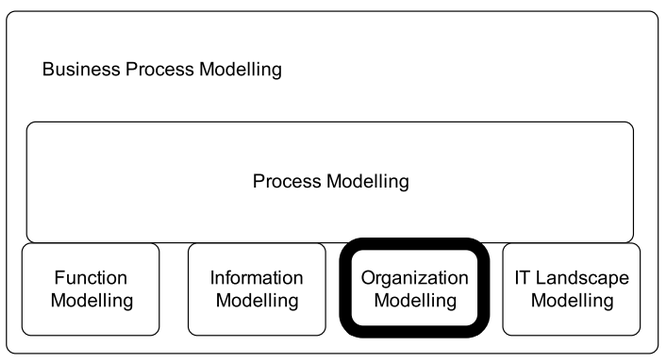


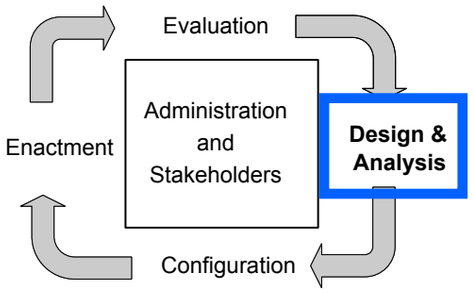
Organizational models

Organizational structure must be represented

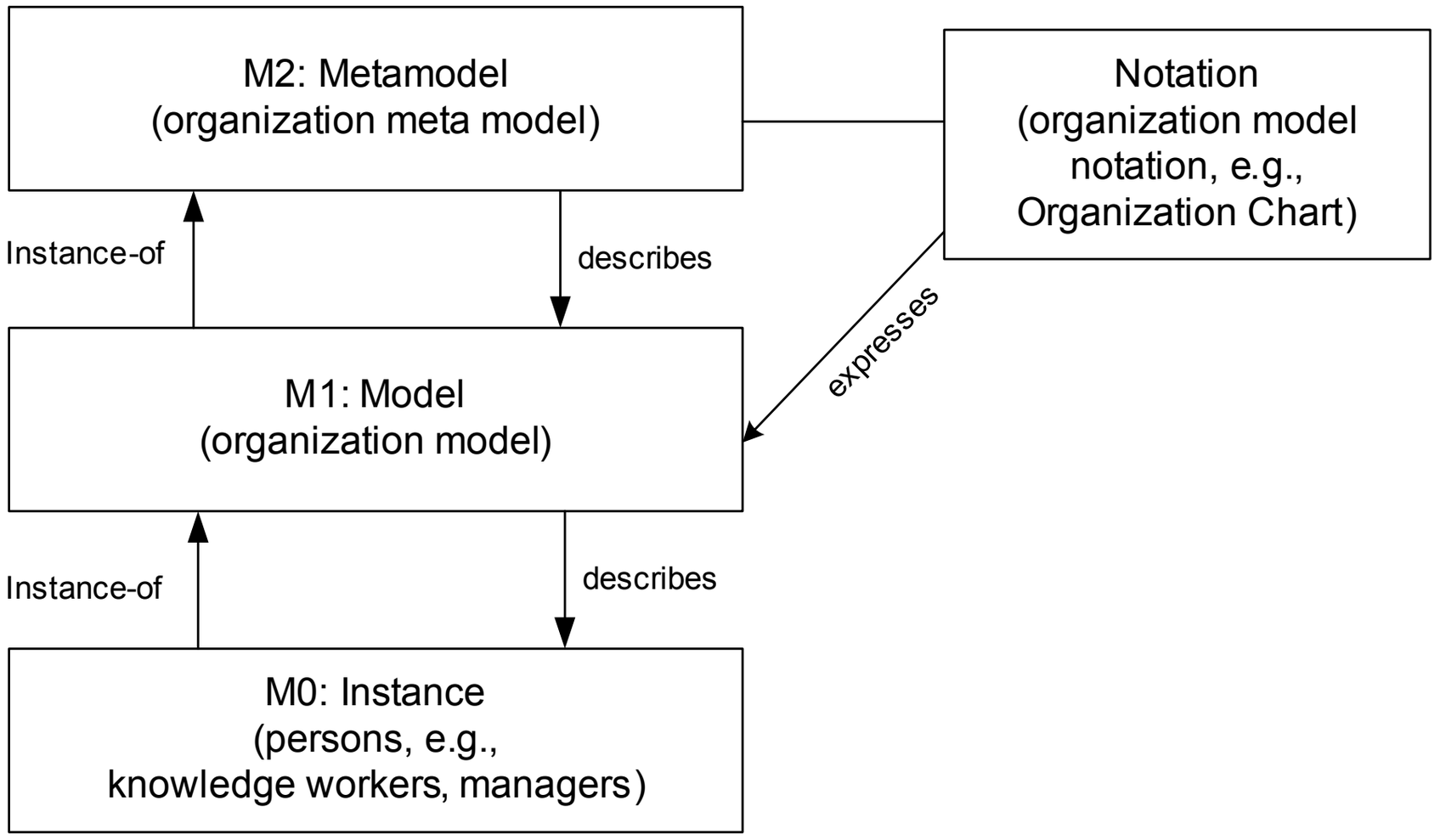
Activities must be associated

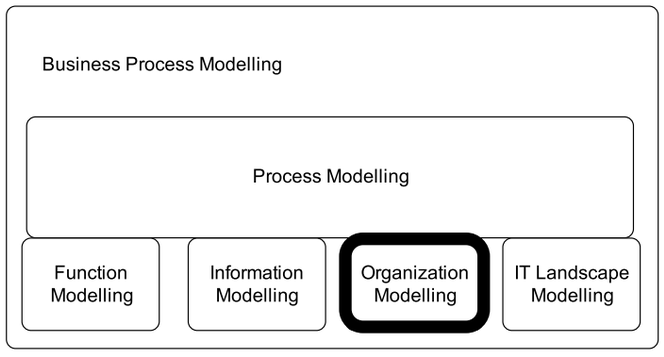
to specific roles or departments

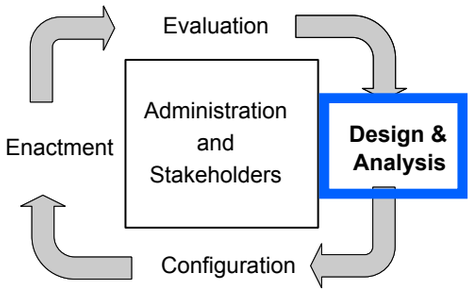




Organizational models



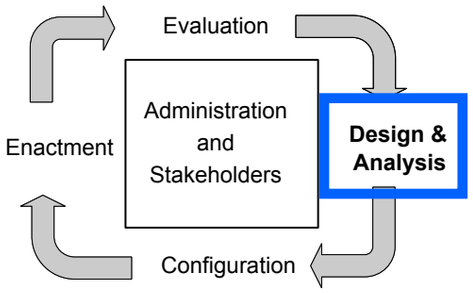




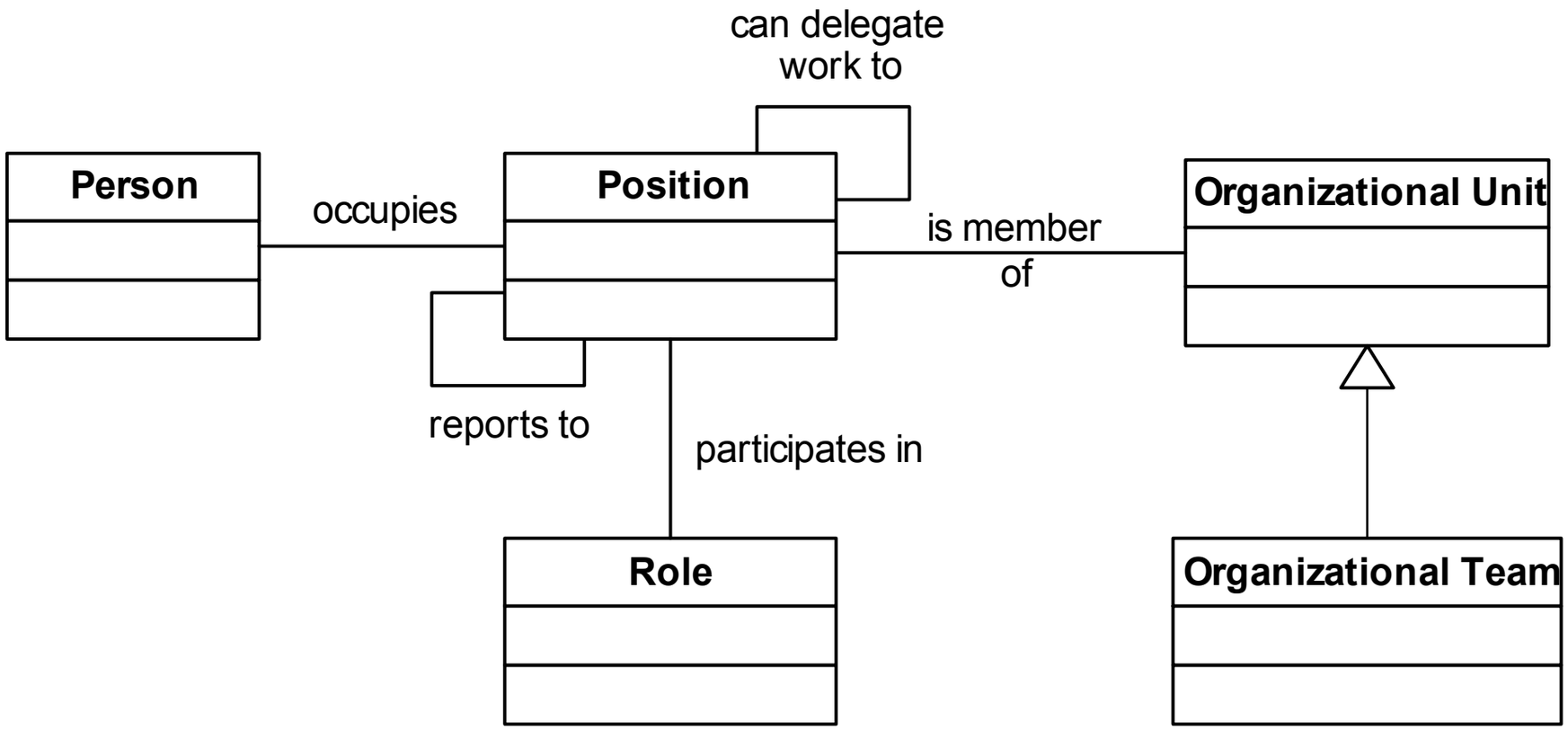
Roles

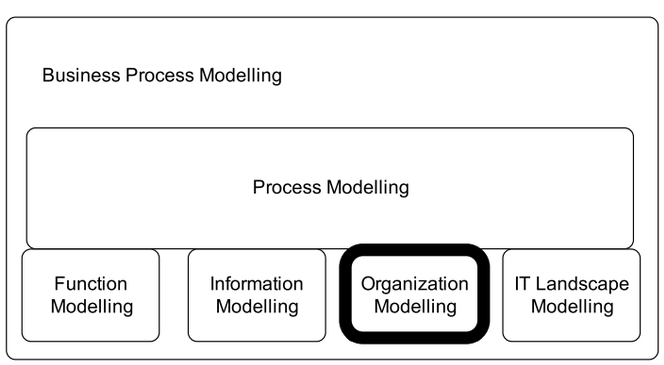
Roles are groups of employees that qualify for being responsible of certain activities.

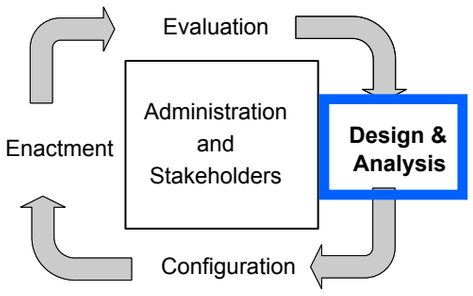
Increased flexibility:
different persons can cover the same role at different time in different cases



An organizational metamodel



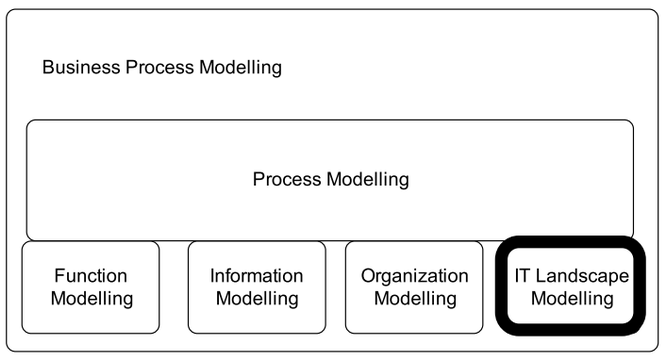


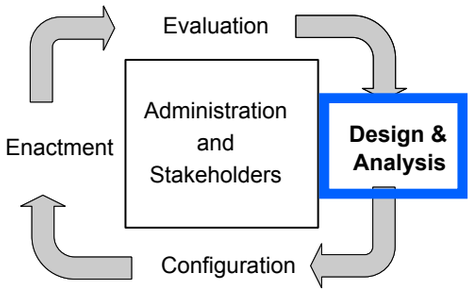


IT landscape

Many activities in a business process are supported by information systems

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



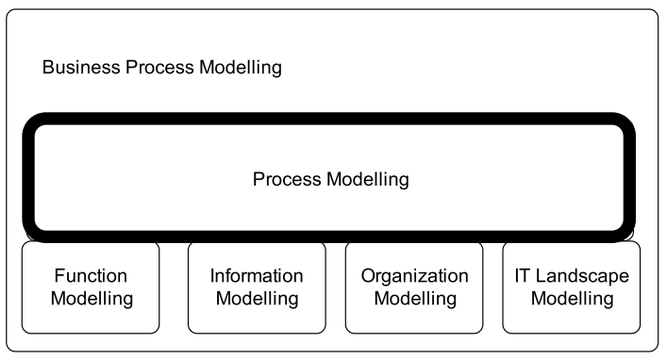


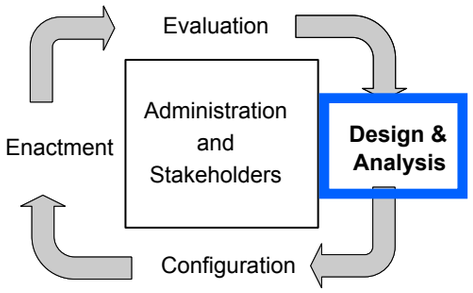
Process models

Define the glue between the subdomains

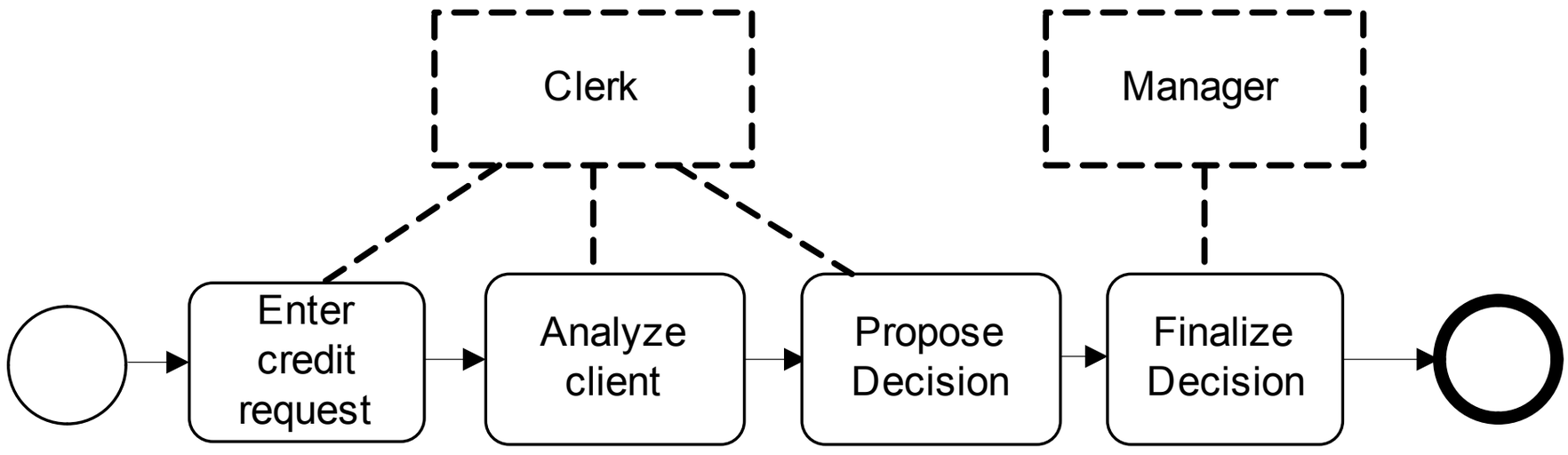
Relate functions and execution constraints

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

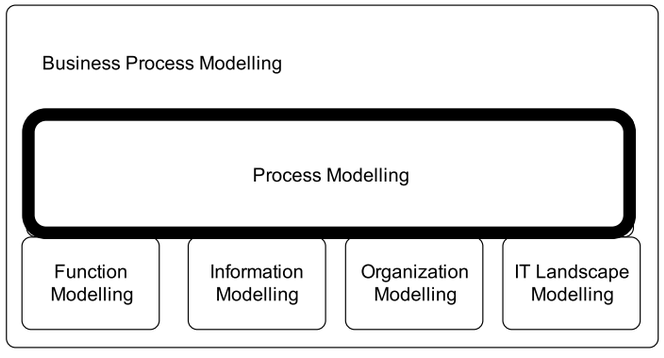


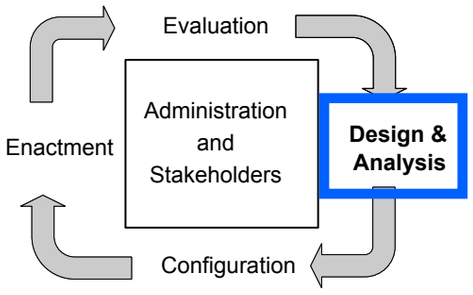


A process model with role information

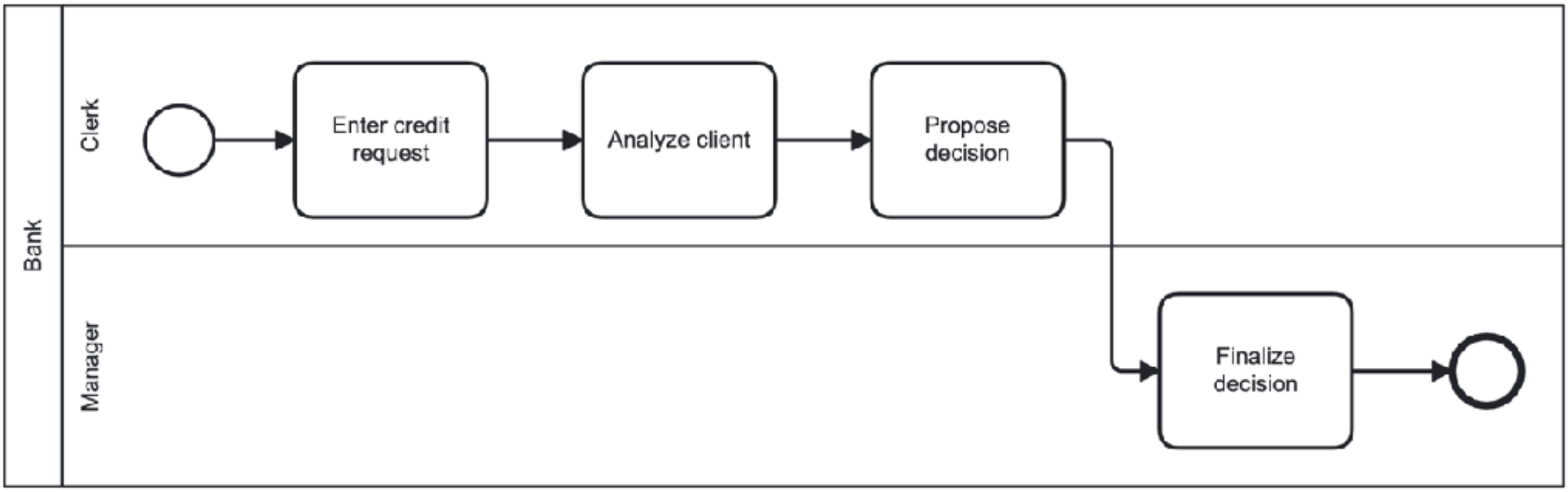


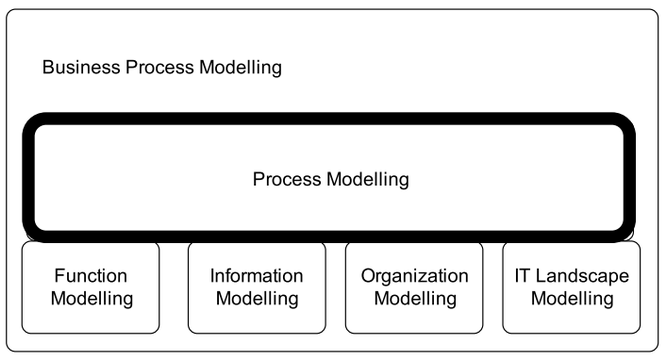
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

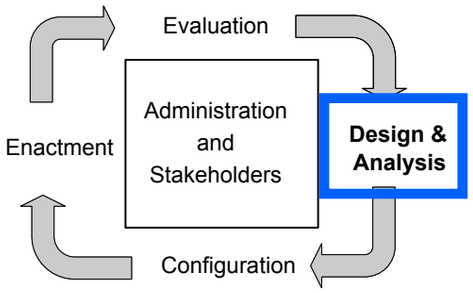




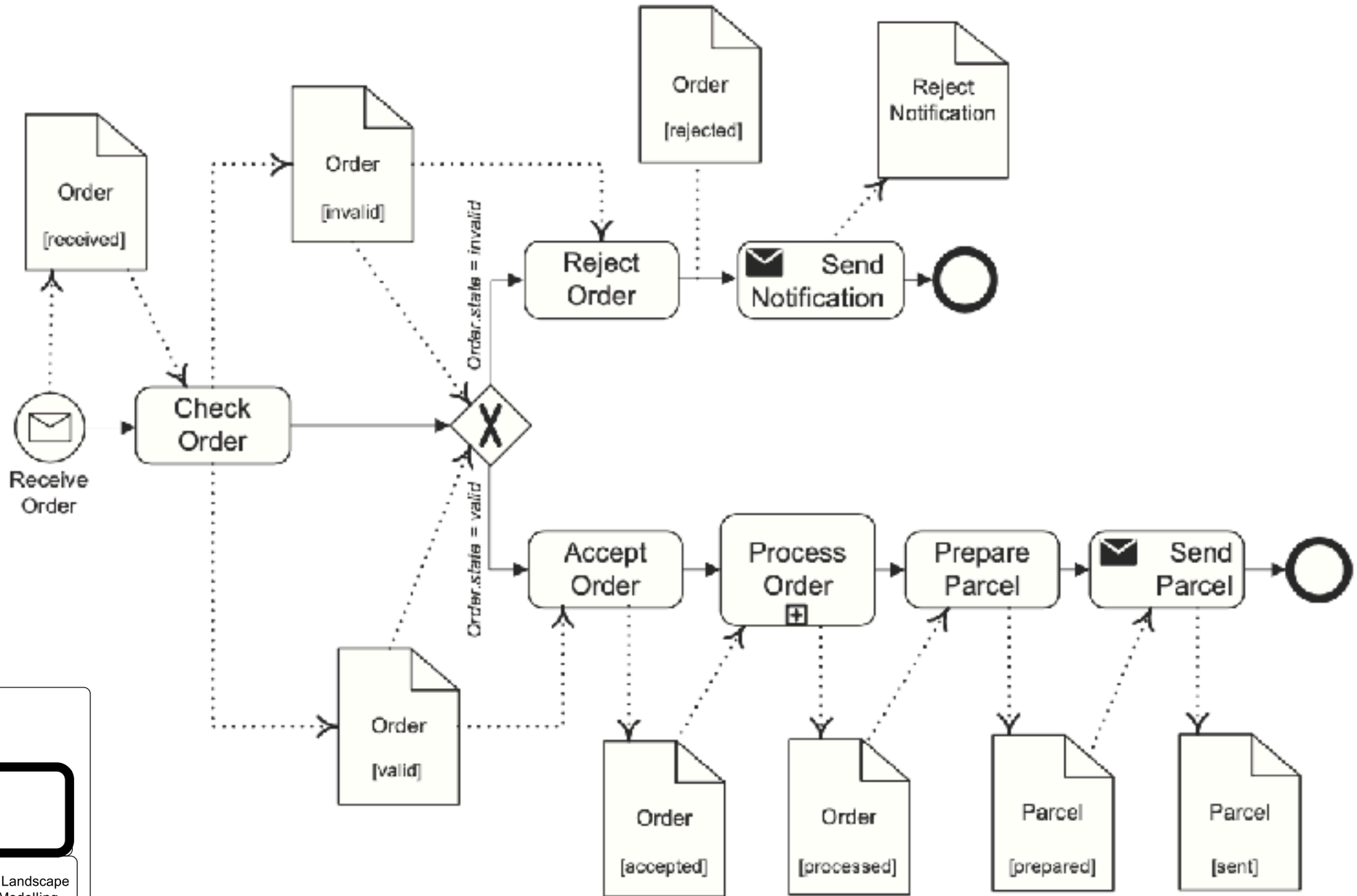
A BPMN model with role information

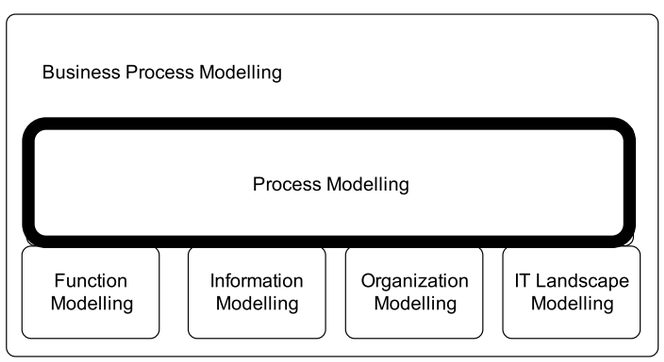


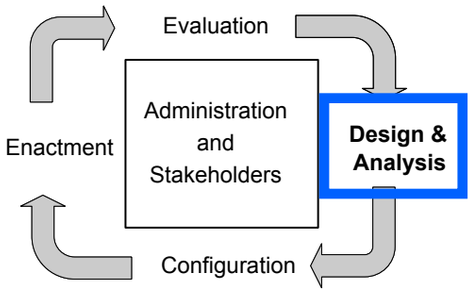




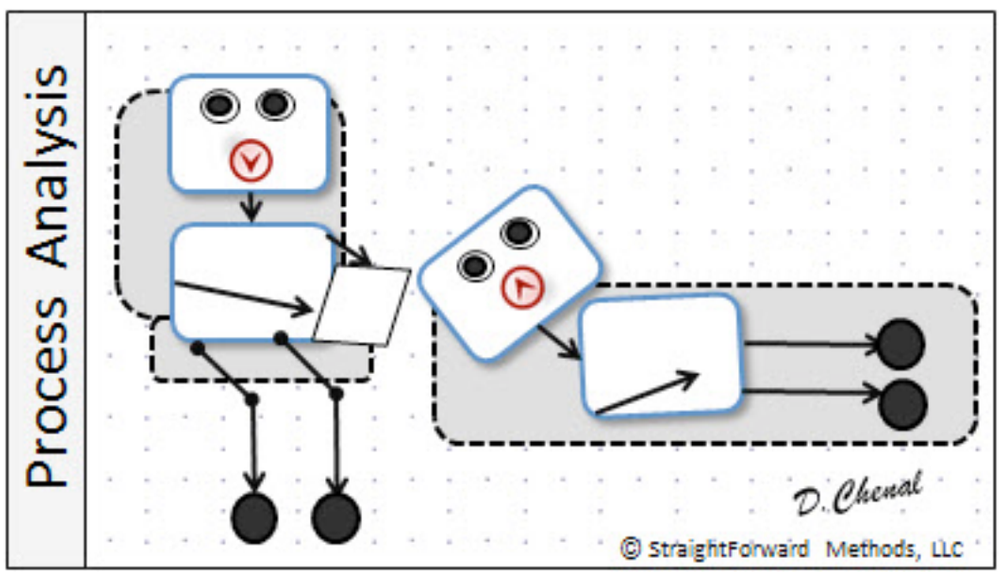
A BPMN model with data objects

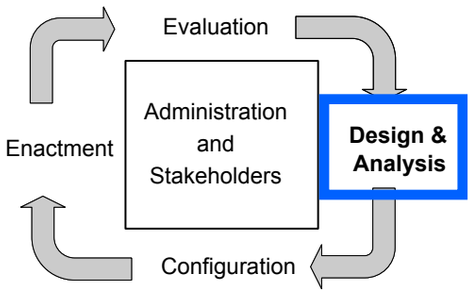





Evaluation
Administration
and
Stakeholders
Design &
Analysis
Enactment
Configuration

Analysis

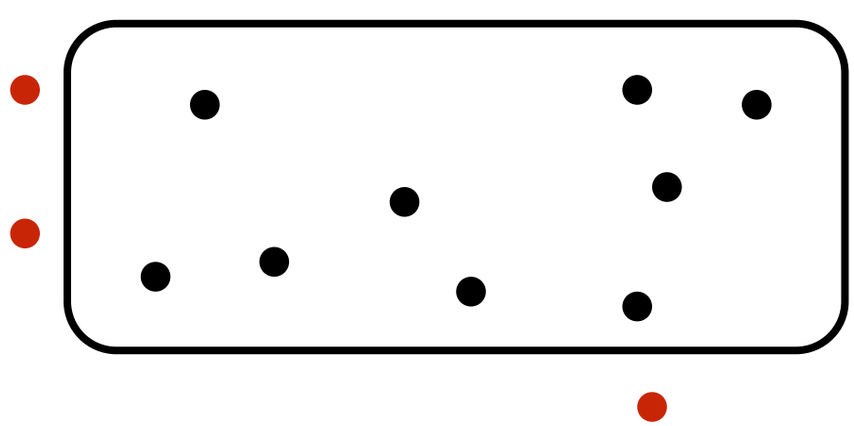

Process Analysis
D.Chenal
© StraightForward Methods, LLC



Analysis: Validation

The initial design must be validated by checking that **all valid process instances are reflected** by the business process model

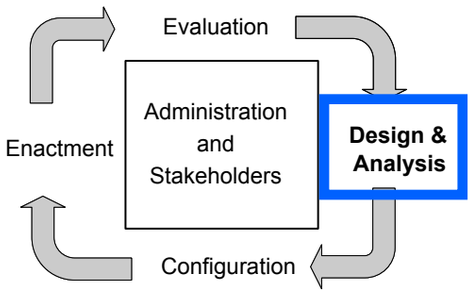
Useful instrument: a **workshop** where the persons involved can discuss the business process model



desired instances

undesired instances

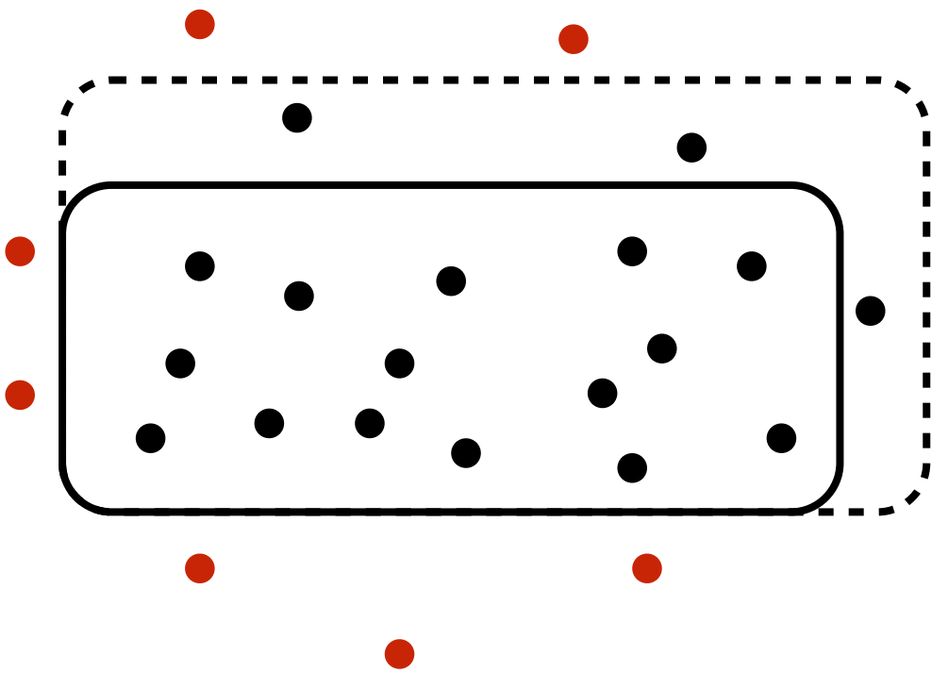
proposed process

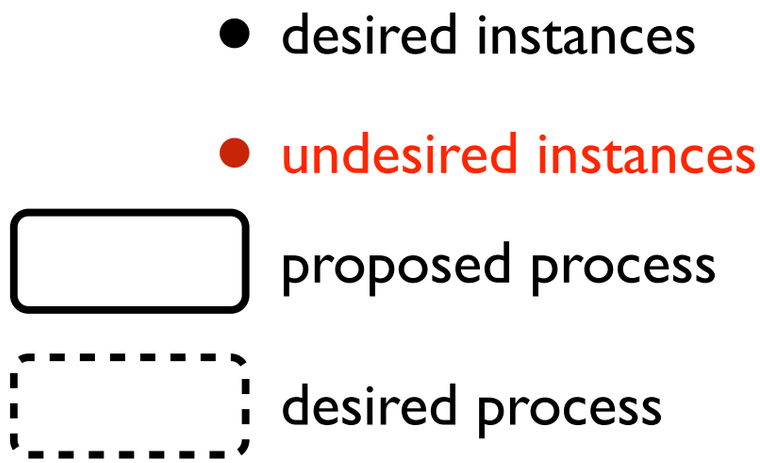


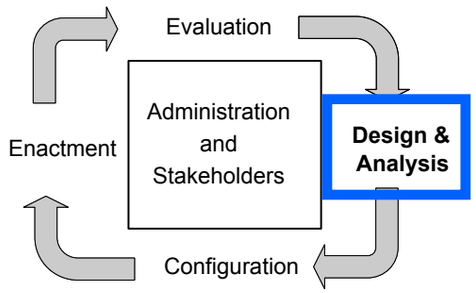
Analysis: Validation

The initial design must be validated by checking that **all valid process instances are reflected** by the business process model

Useful instrument: a **workshop** where the persons involved can discuss the business process model







Analysis: Simulation

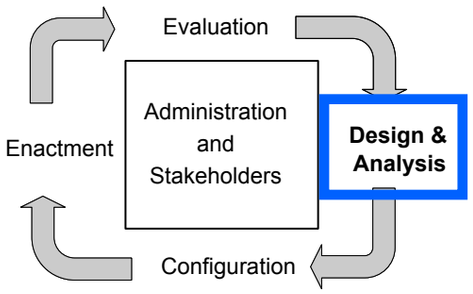
Simulation techniques can support validation

Stakeholders can walk through the process in a step-by-step manner

Check whether processes expose all desired behaviour

Estimate performance measures (e.g., time, cost,…)

Discover undesired execution sequences to show deficits in the process model



Analysis: Verification

The business process model must be analyzed and improved to make sure:

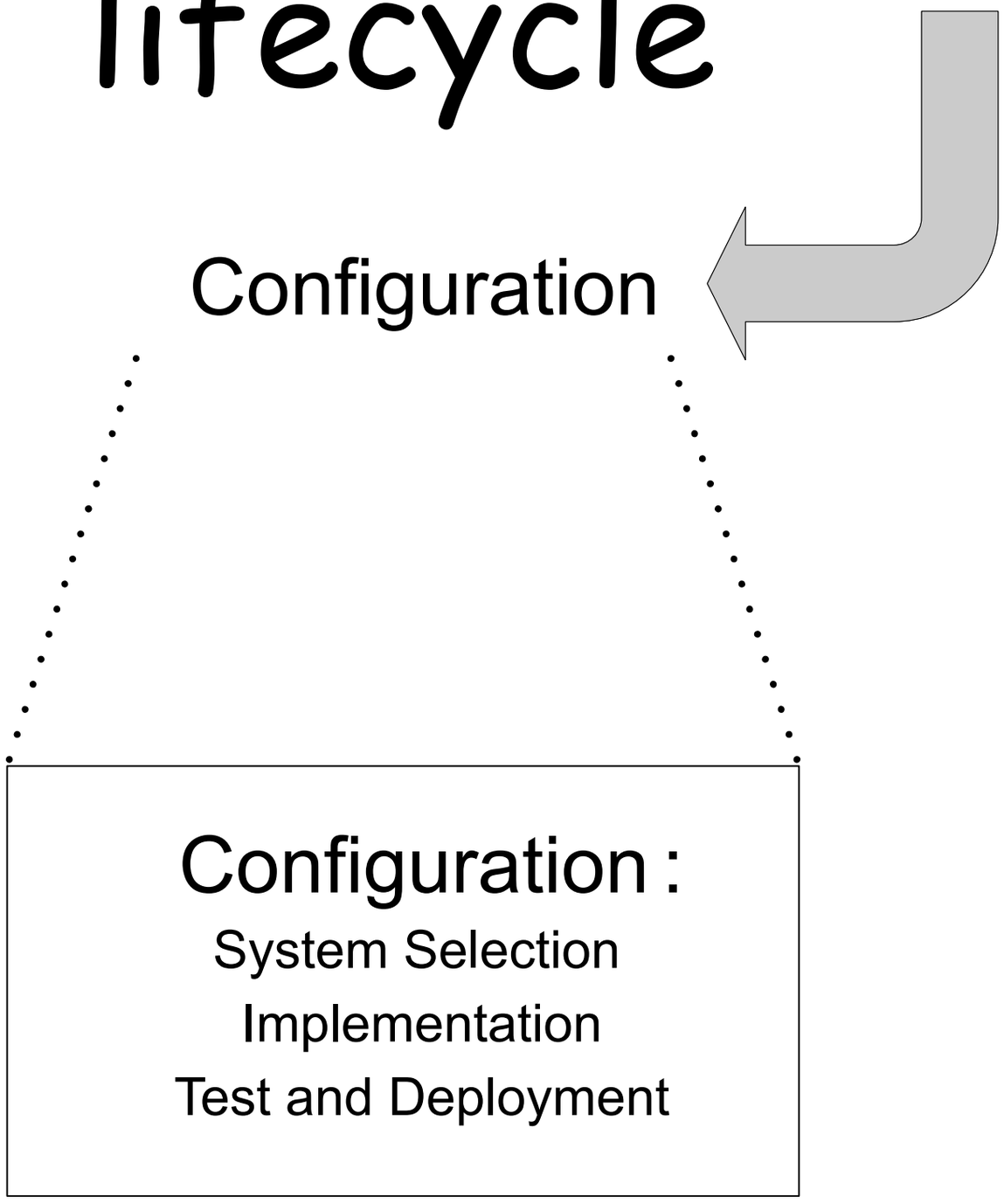
all tasks can be used in some instance

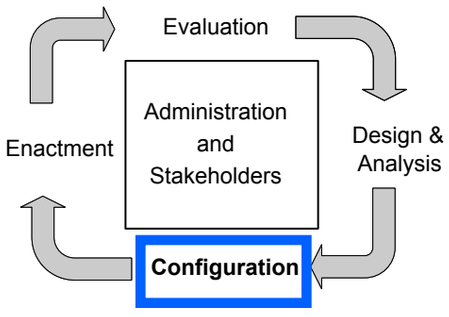
it can always come to an end
(e.g., absence of deadlock)

it actually includes all desired instances
it does not allow any undesired instance

Error-prone activities, to be repeated several times, for which automatic tools are necessary

Business process lifecycle



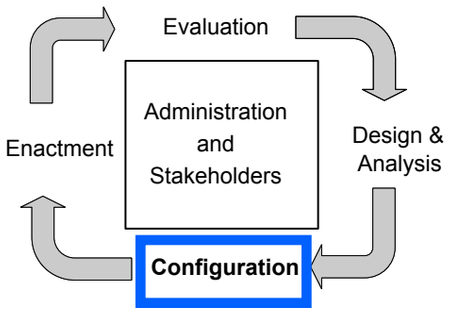


Configuration phase

From (verified) business process models to **implementation** as

a set of policies, guidelines and procedures (to be followed by employees)

a dedicated software system (over a chosen implementation platform: a business process management system)

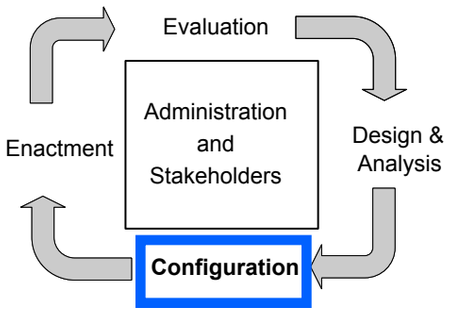


Process-driven software

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)

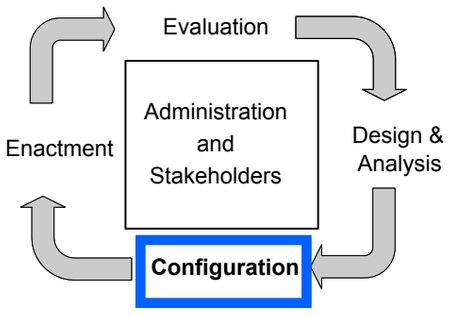


Enhanced models

Software systems usually require additional **technical information**

The model must be decorated with such data, to be exploited for configuring the system

Examples: interactions of the employees with the system, integration of existing systems, wrapping of legacy software



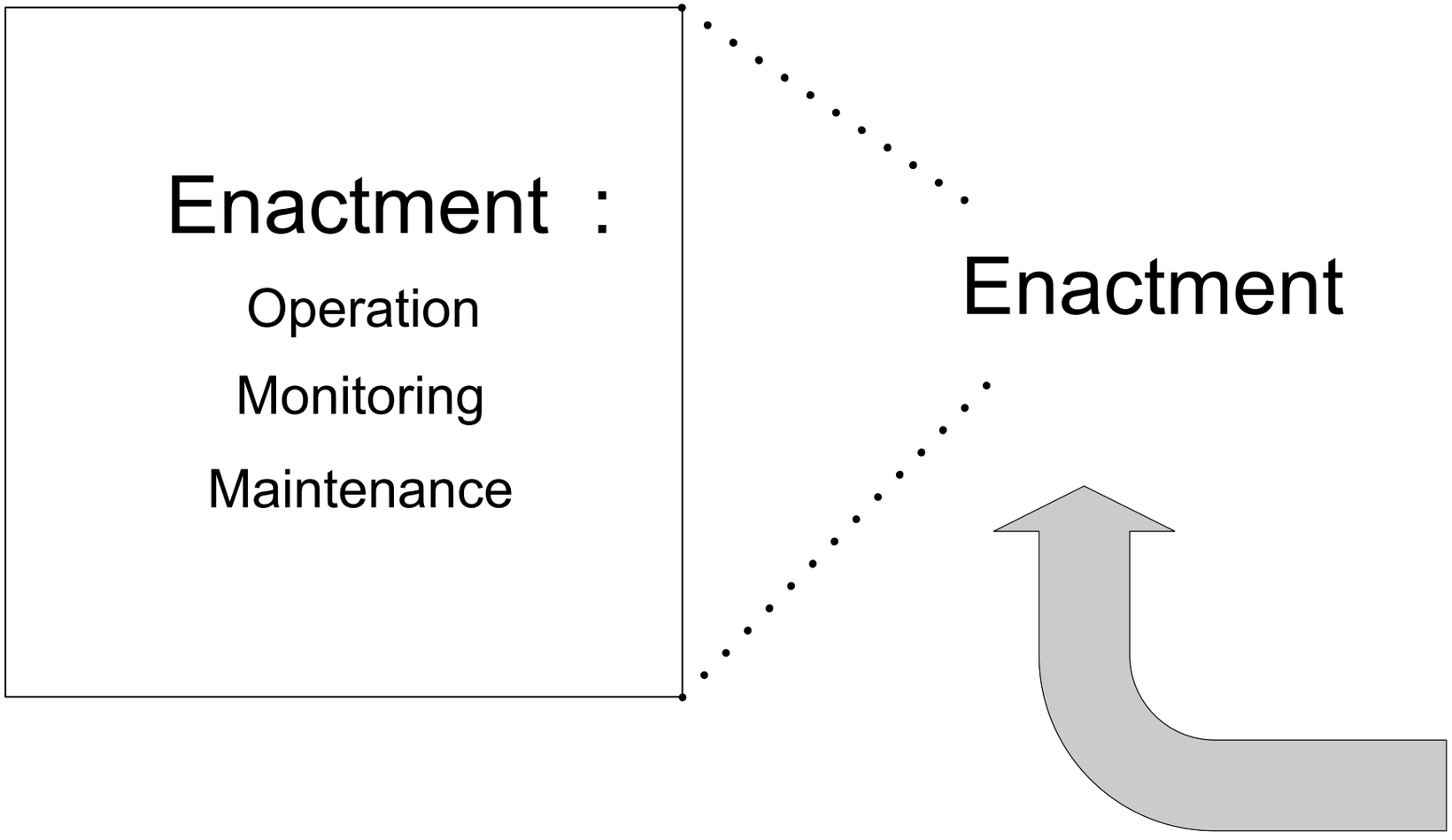
Testing

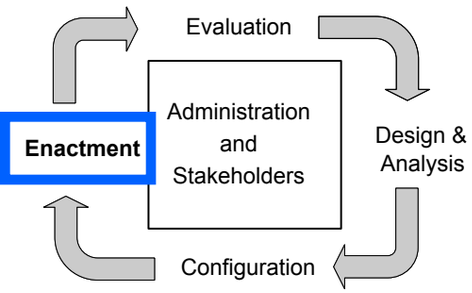
When the system is configured,
it must be tested before deployment

Usual testing techniques from SW engineering
Integration tests
Performance tests

Other possible activities:
training of personnel,
migration of application data

Business process lifecycle



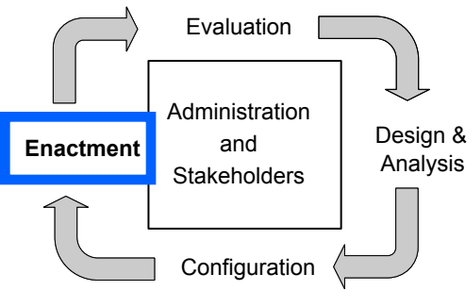


Enactment phase

When the system is deployed,
business process instances can be **enacted**

Typically, each process instance is initiated after
an **event** occurs
(e.g., the receipt of an order)

The system must control and monitor the
execution of all instances according to the model
to guarantee a correct process **orchestration**
(e.g., respecting dependencies)

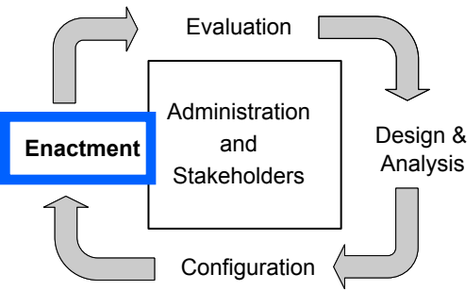


Enactment

Activities can be performed by employees **manually** or by the help of information systems

Other activities can be enacted **automatically** by information systems

Some activities can **trigger** or **inhibit** other activities



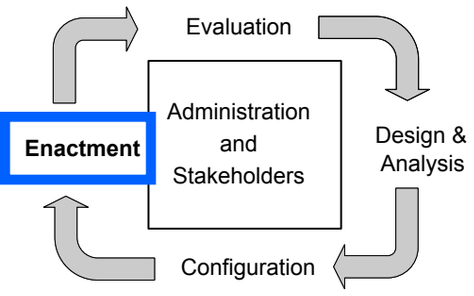
Monitoring

At each moment in time, the current **status** of any instance must be **known** (and **logged**) by the system as accurately as possible

Both for process instances and activity instances

Helpful visualization techniques can be provided by business process management systems (e.g., coloured activities)

Such information is highly valuable for customers (e.g., tracking of orders)



Event Log Example

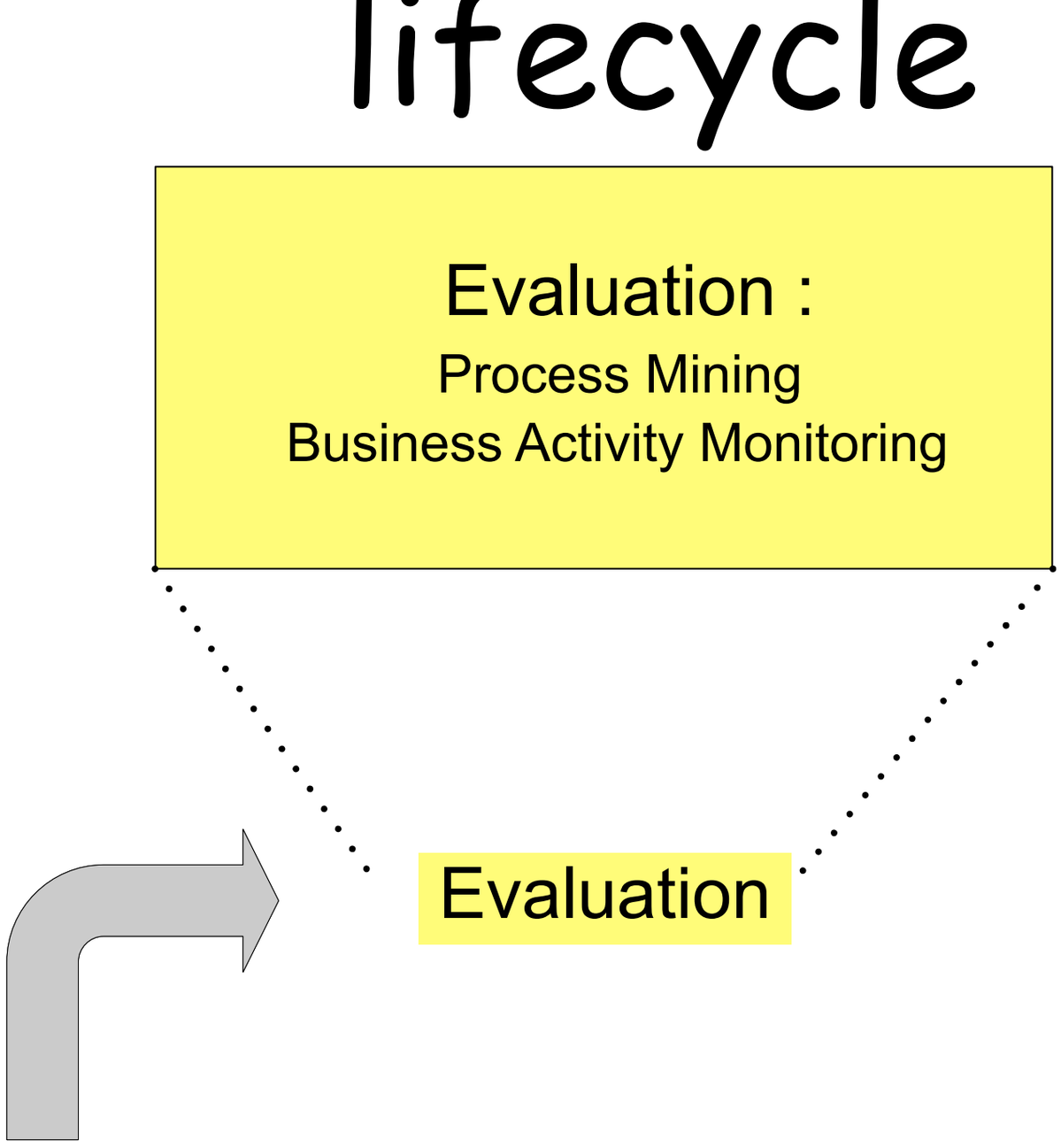
Case id	Event id	Properties				
		Timestamp	Activity	Resource	Cost	...
1	35654423	30-12-2010:11.02	Register request	Pete	50	...
	35654424	31-12-2010:10.06	Examine thoroughly	Sue	400	...
	35654425	05-01-2011:15.12	Check ticket	Mike	100	...
	35654426	06-01-2011:11.18	Decide	Sara	200	...
	35654427	07-01-2011:14.24	Reject request	Pete	200	...
2	35654483	30-12-2010:11.32	Register request	Mike	50	...
	35654485	30-12-2010:12.12	Check ticket	Mike	100	...
	35654487	30-12-2010:14.16	Examine casually	Pete	400	...
	35654488	05-01-2011:11.22	Decide	Sara	200	...
	35654489	08-01-2011:12.05	Pay compensation	Ellen	200	...

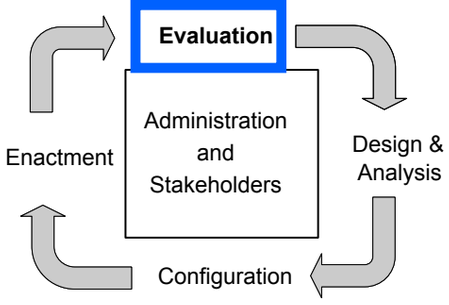
Business process lifecycle

Evaluation :

Process Mining

Business Activity Monitoring



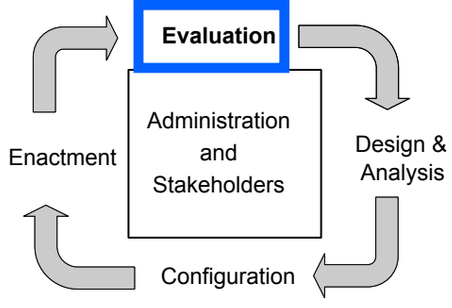


Evaluation phase

Execution **logs** are of fundamental importance

The information collected during instances enactment can be used to evaluate and improve business process models

Business **activity monitoring** and **process mining** techniques aim at identifying the quality of the model and the adequacy of the environment

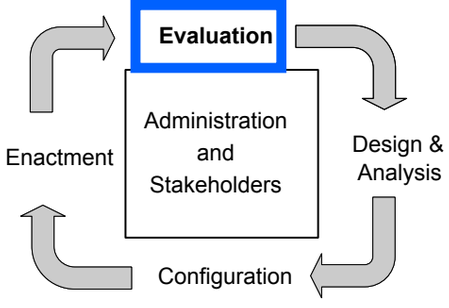


BA monitoring

Log files typically include information such as the start / end timestamps of activity instances

Activity monitoring serve to identify that certain activities take too long or need more resources

The same information can be also exploited in the simulation sub-phase of the design and analysis phase



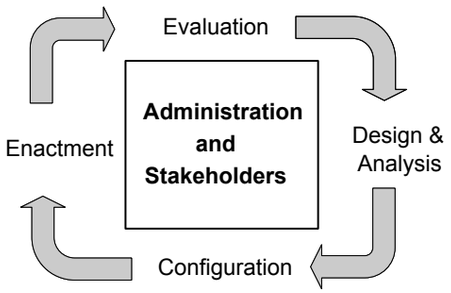
Mining

Process mining has recently turned into an active field of research

Thanks to mining techniques, execution logs can be used for the automatic generation of business process models in the design and analysis phase

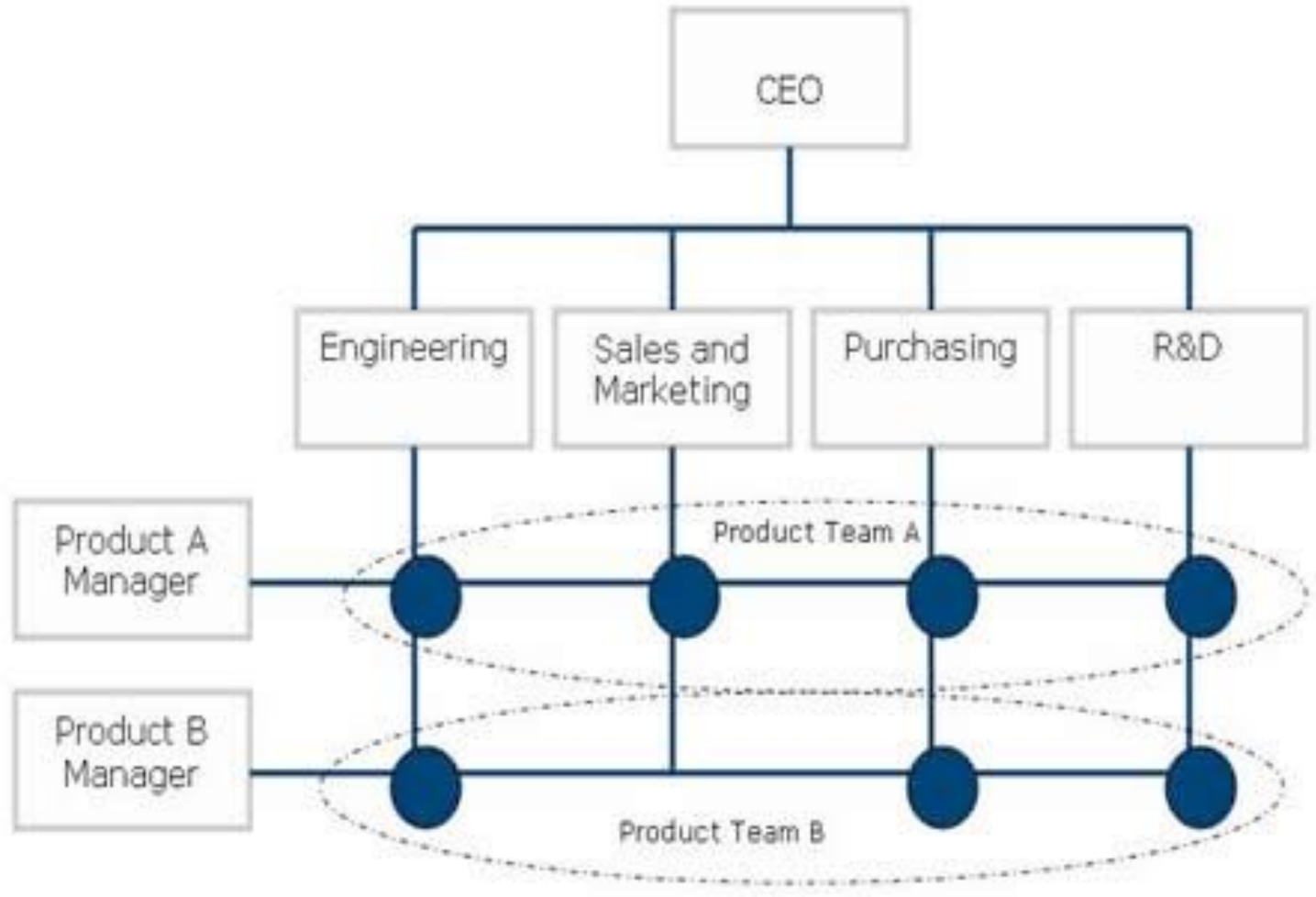
They can also be used to assess and compare different models to see which fits best the enacted instances

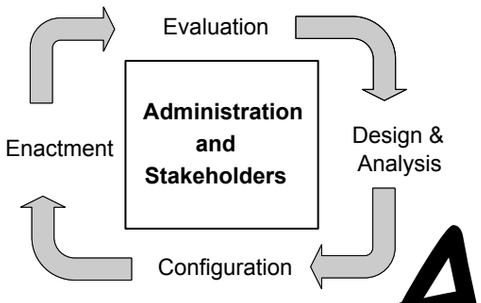
Business process lifecycle



Context

Matrix organizational structure



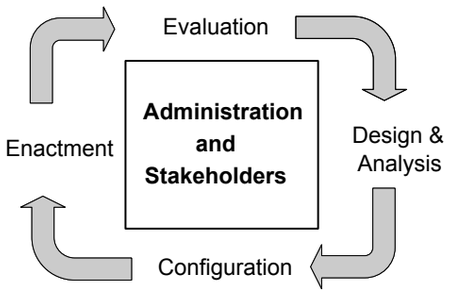


Administration phase

Business process management involves numerous artifacts at different levels of abstraction

Such artifacts need to be organized and managed (storage, retrieval, disposal)

A well-structured repository is needed, with powerful query mechanisms

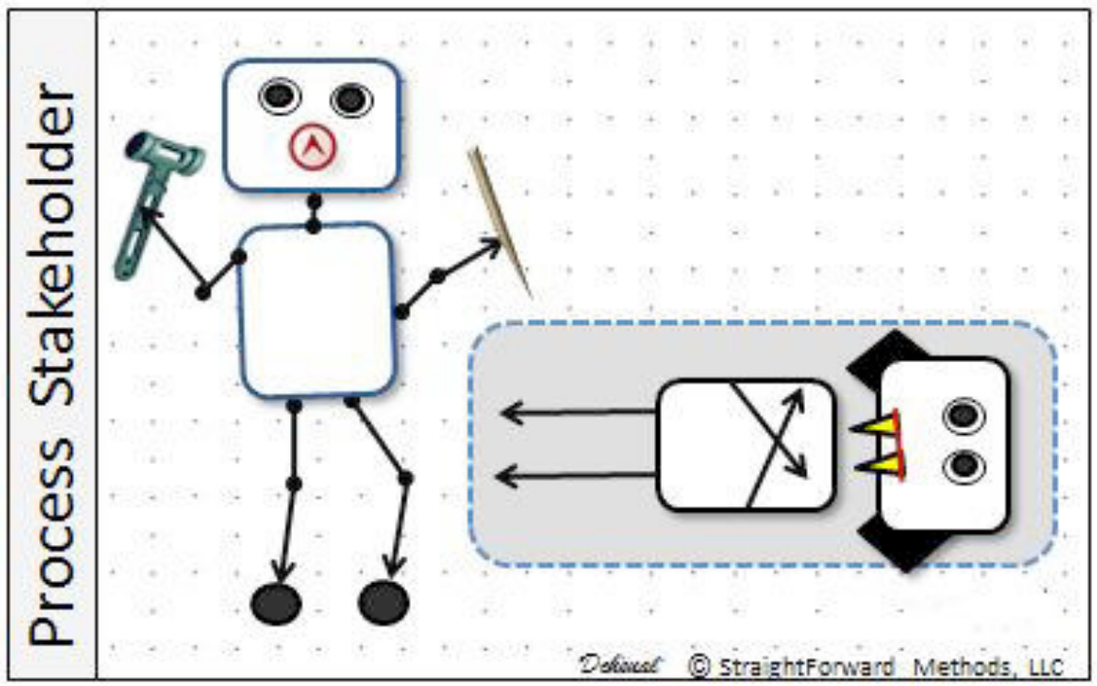


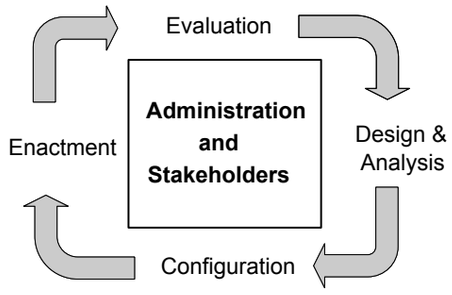
Stakeholders

Several types of stakeholders co-exist in the process domain

They have different kind of educational background, knowledge, expertise, experience

Roughly, they can be classified into a few roles

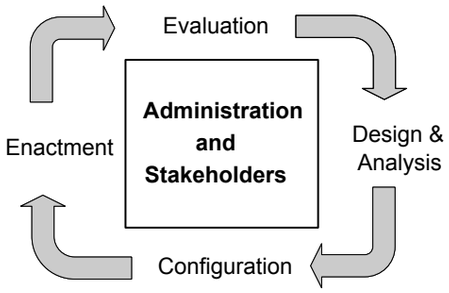




Chief process officer

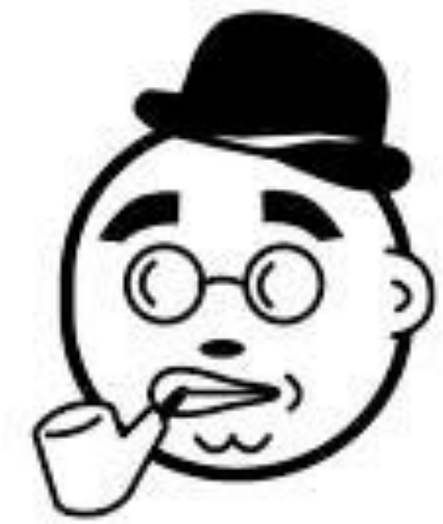
Top level management
(CPO reports directly to CEO / board of directors)

Responsible for defining **rules, policies and guidelines** and for standardizing and harmonizing business processes in the enterprise

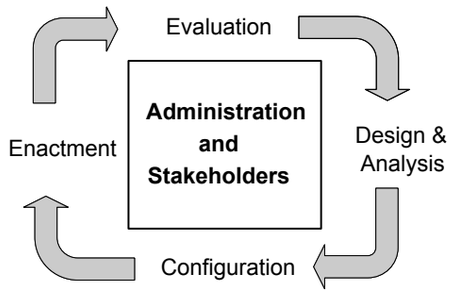


Business engineer

Business domain expert
responsible for defining **strategic goals** of the company and **organizational business processes**



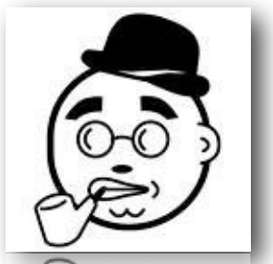
Often equipped with non-technical educational background (mostly economics)
simple-to-use process modeling notation
are the perfect communication mean

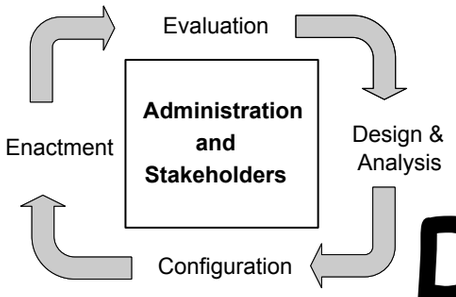


Process designer

Responsible for **modeling business processes** by communicating with business domain experts and other stakeholders

Must be equipped with good analytical capabilities and **excellent communication skills**



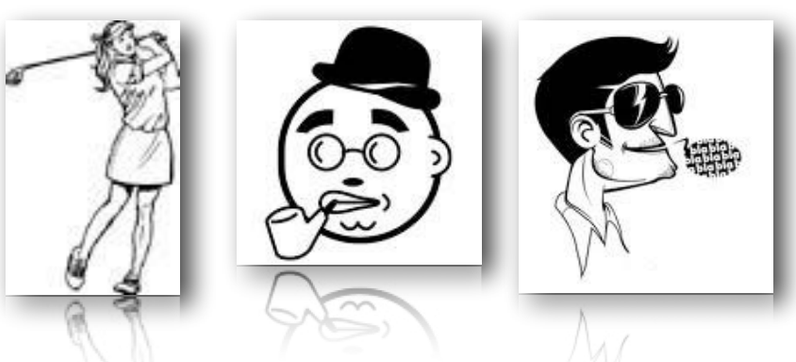


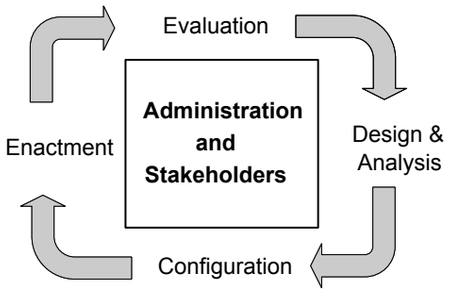
Process participants

Conduct the actual **operational work** during the enactment of processes

They are knowledgeable about the activities conducted, fundamental information for the modeling phase

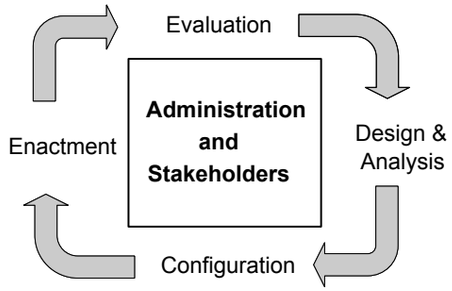
Their information must be assembled by the designer to compose an overall picture in the process model





Knowledge worker

Process participants who use software systems to **perform activities** in a business process, often autonomously

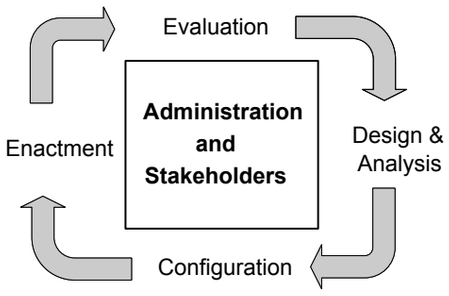


Process responsible

An individual who is held responsible for the correct and efficient execution of all instances of a business process model

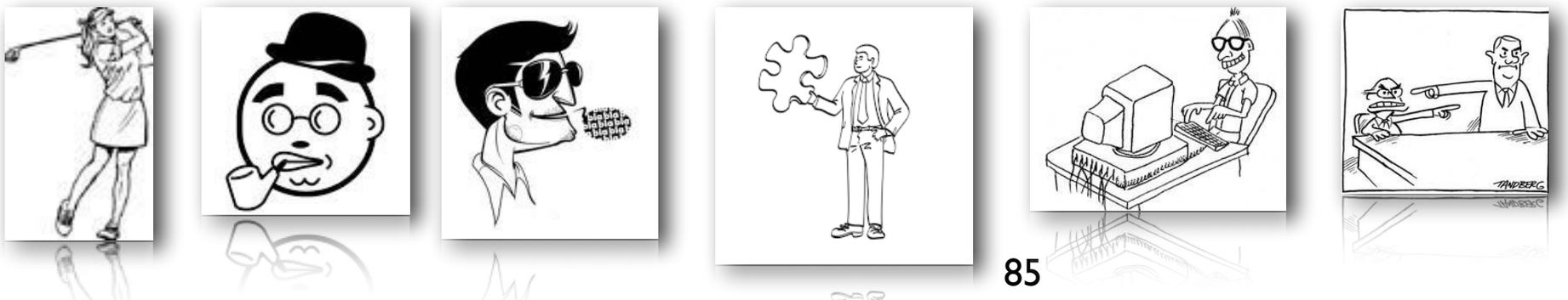
Responsible for **detecting inefficiencies** and **improving** the process model

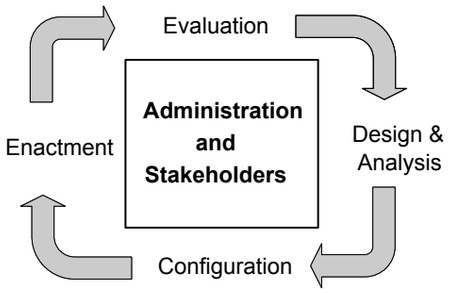
Close collaboration with process participants and the process designer is needed



System architect

Responsible for developing and **configuring** business process management systems on the information system infrastructure at hand



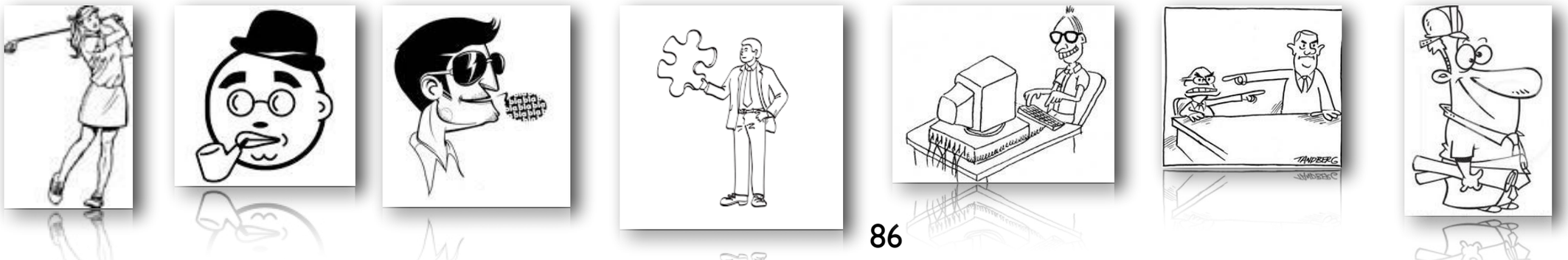


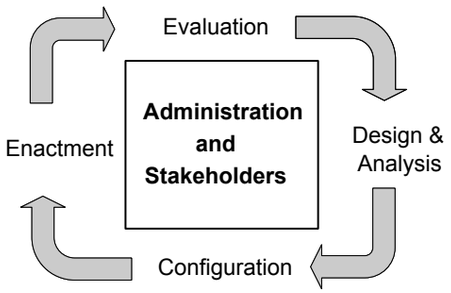
Developers

Information technology professionals

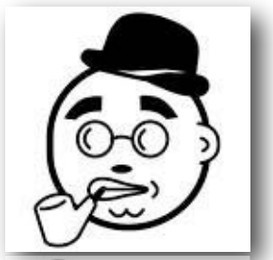
Responsible for creating the **software artifacts** required to implement business processes

Implementation of interfaces is a relevant part of the work done by developers



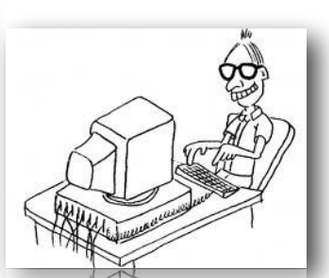


Chief Process Officer: policies and guidelines



Business Engineer: organizational business processes

Process Designer: business processes modeling

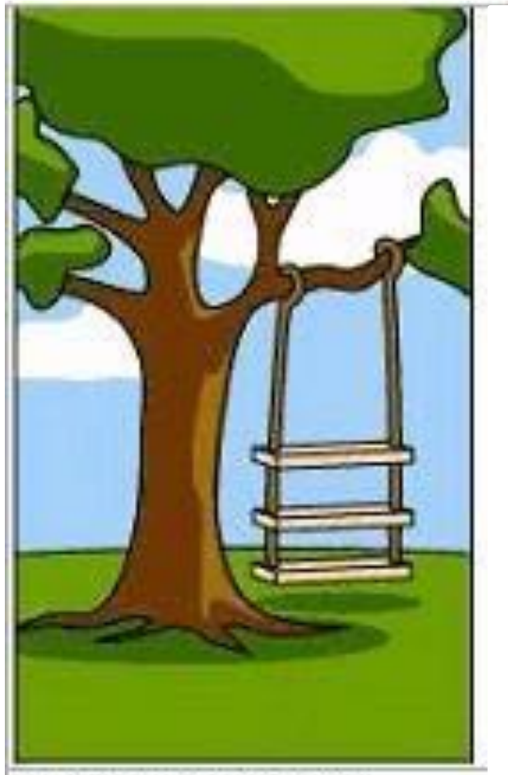


Process Participants / Knowledge Workers: operational work

Process Responsible: monitoring and improvement

System Architect / Developers:
IT infrastructure and SW artifacts configuration

Requirements gone bad



How the customer explained it