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
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 eighth).
Five phases, with logical dependencies, organized along a cyclic structure:

- **Enactment**
  - Operation
  - Monitoring
  - Maintenance

- **Configuration**
  - System Selection
  - Implementation
  - Test and Deployment

- **Design & Analysis**
  - Business Process Identification and Modelling

- **Evaluation**
  - Process Mining
  - Business Activity Monitoring

- **Administration and Stakeholders**

**BP lifecycle**
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

- **Plan**: Fix objectives and how to achieve them
- **Do**: Execute plan
- **Check**: Measure results and compare them to expected goals
- **Act**: Detect issues and adjust goals
Business process lifecycle

Design & Analysis

Design:
Business Process Identification and Modeling

Analysis:
Validation
Simulation
Verification
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)
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

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.
Nella seconda fase si è proceduto con la trasformazione dei diagrammi BPMN in Petri nets, quindi con l’analisi di quest’ultime. Le reti sono state costruite utilizzando il tool WoPeD ed applicando fedelmente le regole di conversione da BPMN a Petri nets.

Analisi semantica e grafo di raggiungibilità

Dall’analisi semantica eseguita mediante WoPeD, la rete, contenente 36 piazze e 40 transizioni, risulta essere:

- free-choice, cioè ogni volta che c’è un arco \((p, t)\), allora c’è un arco da ogni piazza di input di \(t\), ad ogni transizione di output di \(p\);
- s-coverable, ogni piazza, infatti, appartiene all’unico \(s\)-component presente; da qui si può inoltre dedurre che è possibile derivare \(s\)-invariants positivi.
- well structured, caratteristica garantita dall’assenza di TP-handles e PT-handles (in particolare nessuno dei flussi alternativi creati mediante i diversi XOR-splits risulta sincronizzato mediante AND-joins);
- workflow net, ovvero è presente un’unica piazza iniziale \(\text{start}\) con \(\text{start} = 0\), un’unica piazza finale \(\text{end}\) con \(\text{end} = 0\) e tutte le altre piazze e transizioni appartengono al cammino da \(\text{start}\) ad \(\text{end}\);
- sound, la rete risulta soddisfare i seguenti vincoli:
  - non sono presenti dead tasks (né dead transitions, né non-live transitions)
  - la piazza finale è raggiungibile da qualsiasi marking \(M\) (option to complete)
  - quando un token raggiunge la piazza finale, tutte le altre piazze sono vuote (proper completion).

Trattandosi di una workflow net, free-choice e sound, un'altra caratteristica garantita è la safeness. Infine, poiché il marking iniziale non è un home marking (cioè non può essere raggiunto da qualsiasi marking), la rete non è cyclic.
Do you know XML?

**eXtensible Markup Language:**
file format for storing and transmitting data

XML tags represent the data structure and contain metadata

```xml
<?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>
```
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
- M3: Meta-Metamodel

Concepts that discipline model definition
- M2: Metamodel

Classes of similar instances
- M1: Model

Concrete entities
- M0: Instance

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

(instance-of)

describes

(better be read bottom-up)

An example:

MOF metamodel (OMG)
Process models and process instances

M2: Metamodel (process meta model)

M1: Model (process model)

M0: Instance (process instance)

Notation (process notation)

Instance-of

describes

expresses
Do you know UML?

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

*Altri dettagli*

- Yes: 15
- Just a little: 17
- No: 16

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)

Each node is associated with at least one edge. The different types of nodes are represented by the generalization relation. Activity models reflect the work units to be performed, event models represent the occurrence of states relevant for the business process, and gateway models represent execution constraints of activities, such as split and join nodes.

While the association between nodes and edges are defined at the node level, the cardinality of the association between special types of nodes (activity models, event models, and gateway models) differs. Each activity model has exactly one incoming and one outgoing edge.

Each process starts with exactly one event, the initial event, and ends with exactly one event, the final event. Therefore, certain events can have no incoming edges (initial event) or no outgoing edges (final event). Gateway models represent control flow. Therefore, they can act as either split nodes or join nodes, but not both. Hence, each gateway model can have multiple outgoing edges (split gateway node) or multiple incoming edges (join gateway node).

Figure 3.14 shows a process model based on the process metamodel introduced. The notation used to express this process model is taken from the Business Process Model and Notation:

- Event model nodes are represented by circles; the final event model is represented by a bold circle.
- Activity models are represented by rectangles with rounded edges.
- Gateway models are represented by diamonds.
- Edges are represented by directed edges between nodes.
A process model

Event
(activity)
(activity)
(activity)
(activity)
Event
(node)
(node)
(node)
(node)
(node)
(node)
Process models and
process instances
Some process instances

1. Enter credit request → Analyze client → Propose Decision → Finalize Decision

2. Enter c.r. (r017, Miller, 10000) → Analyze client (r017, Miller) → Propose Decision (r017, Miller) → Finalize Decision (r017, Miller)

3. Enter c.r. (r018, Brown, 15500) → Analyze client (r018, Brown) → Propose Decision → Finalize Decision

4. Enter c.r. (r019, McGraf, 12000) → Analyze client (r019, McGraf) → Propose Decision (r019, McGraf) → Finalize Decision (r019, McGraf)

5. Enter c.r. (r020, Carey, 20000)
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

- OrderManagement
  - GetOrder
  - CheckOrder
    - AnalyzeOrder
    - SimpleCheck
    - AdvCheck
Modelling:
Vertical Abstraction
Guiding principle

Separation of Concerns (SoC)
(to separate a system into distinct features that overlap in functionality as little as possible)
Look for EWD447: On the role of scientific thought

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

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
<!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>
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
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)

M1: Model
(data model)

M0: Instance
(data values)

Notation
(data model notation, e.g., ER Notation, UML Class Diagrams)

Instance-of

describes

expresses

Business Process Modelling

Process Modelling

Function Modelling
Information Modelling
Organization Modelling
IT Landscape Modelling
Organizational structure must be represented

Activities must be associated to specific roles or departments
Organizational models

M2: Metamodel (organization meta model)

M1: Model (organization model)

M0: Instance (persons, e.g., knowledge workers, managers)

Notation (organization model notation, e.g., Organization Chart)

Instance-of

expresses

describes

Instance-of

Configuration

Enactment

Design & Analysis

Administration and Stakeholders

Evaluation

Business Process Modelling

Function Modelling

Information Modelling

Process Modelling

Organization Modelling

IT Landscape Modelling

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
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

Enter credit request → Analyze client → Propose Decision → Finalize Decision

Clerk

Manager

A BPMN model with role information
A BPMN model with data objects
Analysis
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.
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

Configuration:
System Selection
Implementation
Test and Deployment
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
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:
- Operation
- Monitoring
- Maintenance
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
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

<table>
<thead>
<tr>
<th>Case id</th>
<th>Event id</th>
<th>Properties</th>
<th>Resource</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Timestamp</td>
<td>Activity</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Resource</td>
<td>Cost</td>
</tr>
<tr>
<td>1</td>
<td>35654423</td>
<td>30-12-2010:11.02</td>
<td>Register request</td>
<td>Pete</td>
</tr>
<tr>
<td></td>
<td>35654424</td>
<td>31-12-2010:10.06</td>
<td>Examine thoroughly</td>
<td>Sue</td>
</tr>
<tr>
<td></td>
<td>35654425</td>
<td>05-01-2011:15.12</td>
<td>Check ticket</td>
<td>Mike</td>
</tr>
<tr>
<td></td>
<td>35654426</td>
<td>06-01-2011:11.18</td>
<td>Decide</td>
<td>Sara</td>
</tr>
<tr>
<td></td>
<td>35654427</td>
<td>07-01-2011:14.24</td>
<td>Reject request</td>
<td>Pete</td>
</tr>
<tr>
<td>2</td>
<td>35654483</td>
<td>30-12-2010:11.32</td>
<td>Register request</td>
<td>Mike</td>
</tr>
<tr>
<td></td>
<td>35654485</td>
<td>30-12-2010:12.12</td>
<td>Check ticket</td>
<td>Mike</td>
</tr>
<tr>
<td></td>
<td>35654487</td>
<td>30-12-2010:14.16</td>
<td>Examine casually</td>
<td>Pete</td>
</tr>
<tr>
<td></td>
<td>35654488</td>
<td>05-01-2011:11.22</td>
<td>Decide</td>
<td>Sara</td>
</tr>
<tr>
<td></td>
<td>35654489</td>
<td>08-01-2011:12.05</td>
<td>Pay compensation</td>
<td>Ellen</td>
</tr>
</tbody>
</table>
Business process lifecycle

Evaluation:
- Process Mining
- Business Activity Monitoring

Evaluation
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
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

Administration and Stakeholders
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.
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

Responsibility 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