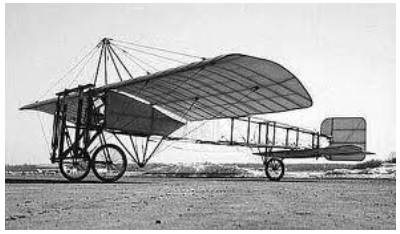
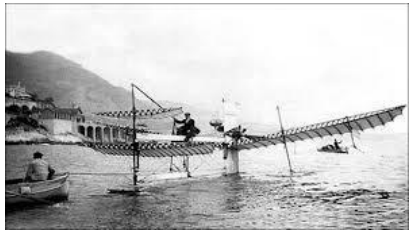
CentraleSupélec

Elements of Complex System Engineering

Antoine B. Rauzy
Department of Mechanical and Production Engineering (MTP)
Norwegian Science and Technology University (NTNU)
and
Chaire Blériot-Fabre
Centrale-Supélec, SAFRAN


CentraleSupélec

LECTURE 1. PART 2.
BUSINESS PROCESS MODEL AND NOTATION

Notions:

- Use Cases
- Business Process Model and Notation

CentraleSupélec

LECTURE 1. PART 2. SECTION 1. INTRODUCTION

CentraleSupélec

Objective of this lecture

To design a system, you need first to answer a number of basic questions: by whom, why and how this system will be operated? When the system is simple, answering these questions are also simple. When it gets complex, things get indeed much more complicated. The best way to elaborate these answers and to share them with stakeholders consists then in developing **use cases**, in other words, to tell "stories" about the system. Use cases are of great help to make things concrete. There exist several modeling languages dedicated to the formalization of uses cases.

In this lecture, we shall start with the analysis of a concrete example: a possible process by which a loan is accepted or rejected. We shall then formalize this example by means of a widely used graphical modeling language, namely the "**Business Process Model and Notation**". Then, we shall provide a textual version of (a small part of) this language and perform some experiments with models designed with this textual version.

CentraleSupélec

Case Study: Loan Assessment Process

A lending agency wish to upgrade its information system. You are called as a IT consultant to help to specify the new system. To do so, you need to understand business processes of this company. One of the key point is to understand how a loan demand is assessed.

Here is what you are explained:

“The sales representative enters the loan demand. The financial analyst assesses the demand. Contracts are managed by the contract department. For each demand, one needs either to create a contract or to update the existing one. The financial analyst must notify the answer to the sales representative so that he or she can inform the client”.

Is that clear? Not really huh?

All right. Let’s model that…

CentraleSupélec

Vocabulary

- A **business process** is a structured collection of **activities**, also called **tasks**, that produces a specific service for one or more given clients. Business processes are often visualized by means of **flow diagrams** (flowcharts) as interleaved sequences of tasks and decision points.
- A **stakeholder** is an individual or collective actor (group or organization) who is actively or passively concerned by a decision or a project; i.e. whose interests can be affected positively or negatively following the execution (or non-execution) of the project.
- A **requirement** is the expression of a documented need on what the system should do or should be. **Functional requirements**, i.e. what the system should do, are usually distinguished from **non-functional requirements** which specify system expected performance levels (in terms of weight, volume, speed…)

CentraleSupélec

LECTURE 1. PART 2. SECTION 2. USE CASES

CentraleSupélec

Use Case

When dealing with problems of complex systems engineering, there is a big risk to reason only from a theoretical point of view: it is quite easy to fell in "**abstract non-sense**". To avoid this pitfall, the best is to rely on concrete examples. Design **use cases**.

A **use case** shows a (typical) way of using the system. It makes it possible to elicit **functional requirements**. Each use case may describe one or more scenarios that show how the system is supposed to interact with users, called actors, to reach a given goal. In a use case, an **actor** can be either a human or an external system.

Modeling formalisms such as **SysML** provide specialized diagrams to describe use cases. It is not clear however that graphical representations help much here. A use case is first and foremost a story, i.e. a text. about the system.

Use Case (example)

Use Case (number 1)

- Name: loan demand assessment;
- Version: 1 ;
- Description: assessment of a demand of loan;
- Actors: sales representative (SR), financial analyst (FA), contract department (CD);
- References: none;
- Prerequisites: the IT system must work correctly;
- Consequents: the demand is accepted or rejected.

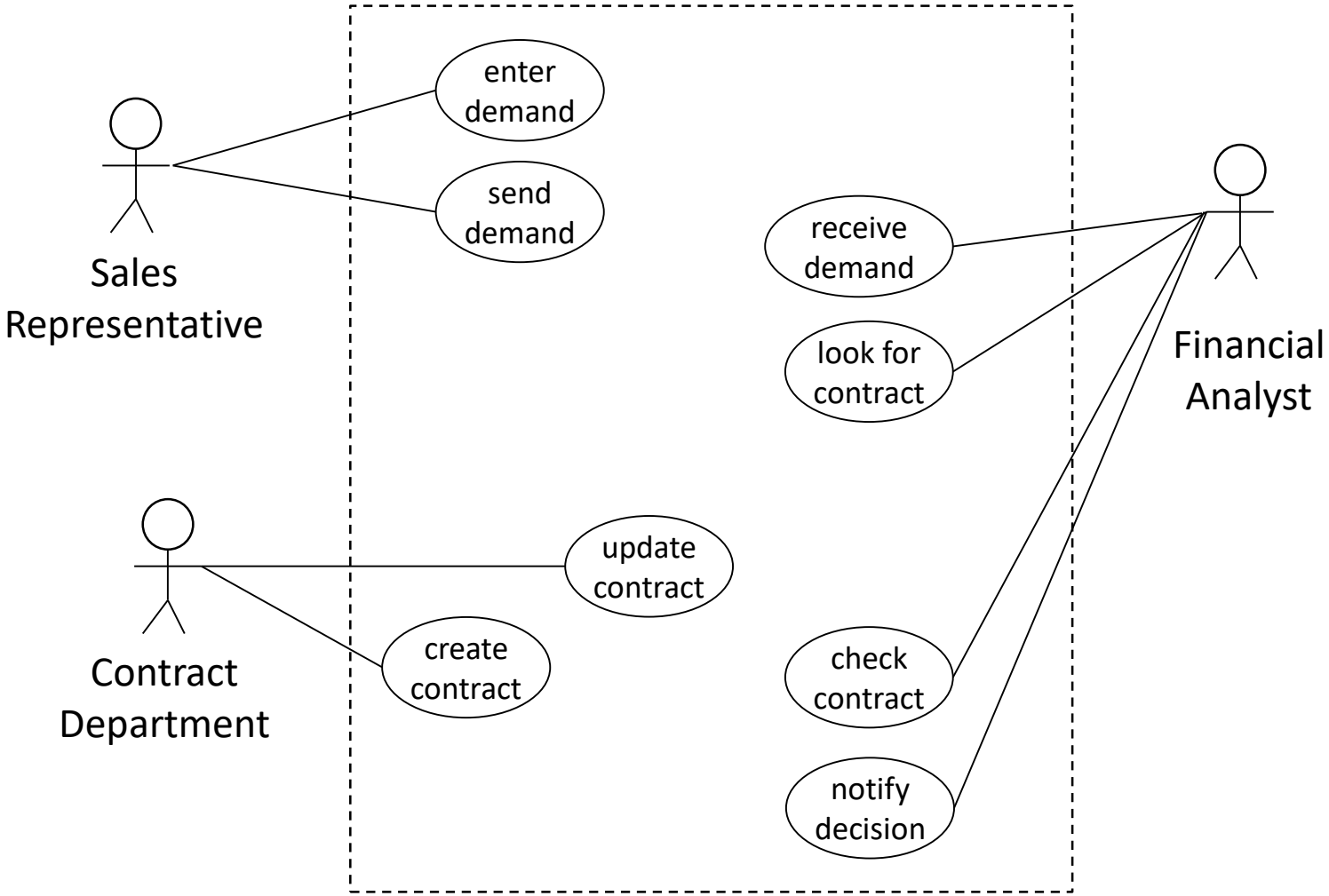
Event sequences

- SR enters the demand in the IT system and sends it to FA.
- FA receives the demand and checks whether it exists a contract in the client's name.
- FA demands to CD to update the contract (if any) or to create it (otherwise).
- CD creates or updates the contract and sends it back to FA.
- FA checks the contract and notifies SR his/her agreement or refusal.

Exceptions

- FA notifies his/her refusal to SR before checking the existence of a contract because the demand cannot be accepted anyway.

Use Case (diagram)



Interest and limits of use cases

Interest:

- Use cases are an efficient mean to **collect requirements** focusing on **interactions** between the **system** and the **actors**.
- Use cases avoid "**abstract non-sense**" and the development of functionalities "for the sake of the elegance".
- Uses cases provided **test scenarios**.

Limits:

- Use cases are too **informal** to be sufficient to specify the system.
- Use cases may hide **business rules and constraints** (technical constraints, regulation, economical constraints…), which may in turn cause problems especially it will be necessary to update the system.

CentraleSupélec

LECTURE 1. PART 2. SECTION 3. BUSINESS PROCESS MODEL AND NOTATION

Business Process Model and Notation (BPMN)

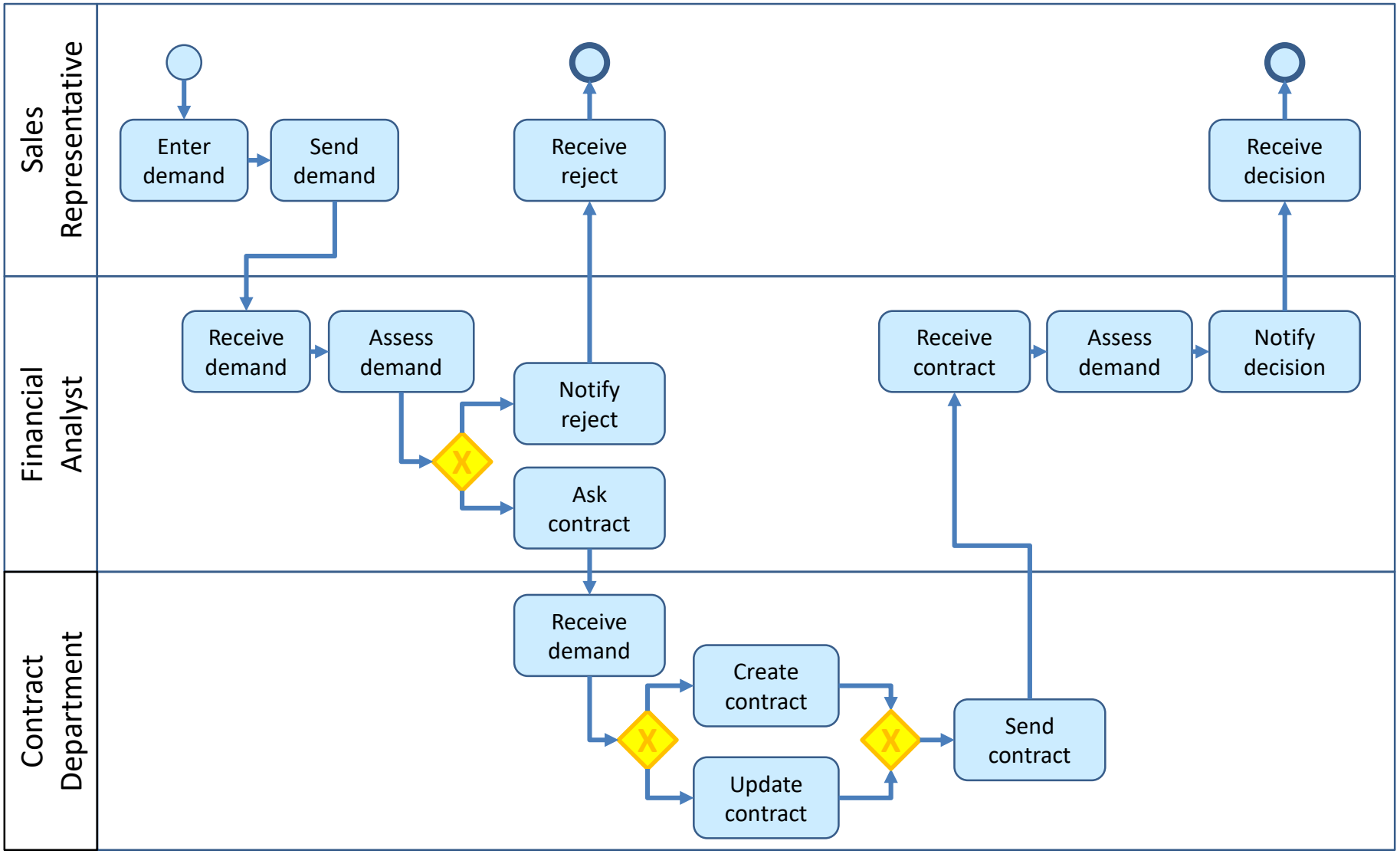
The **Business Process Model and Notation** is a graphical formalism making it possible to describe the different steps of a business process as well as the **exchanges of information** between the **stakeholders** of this process.

Why does BPMN play a very important role?

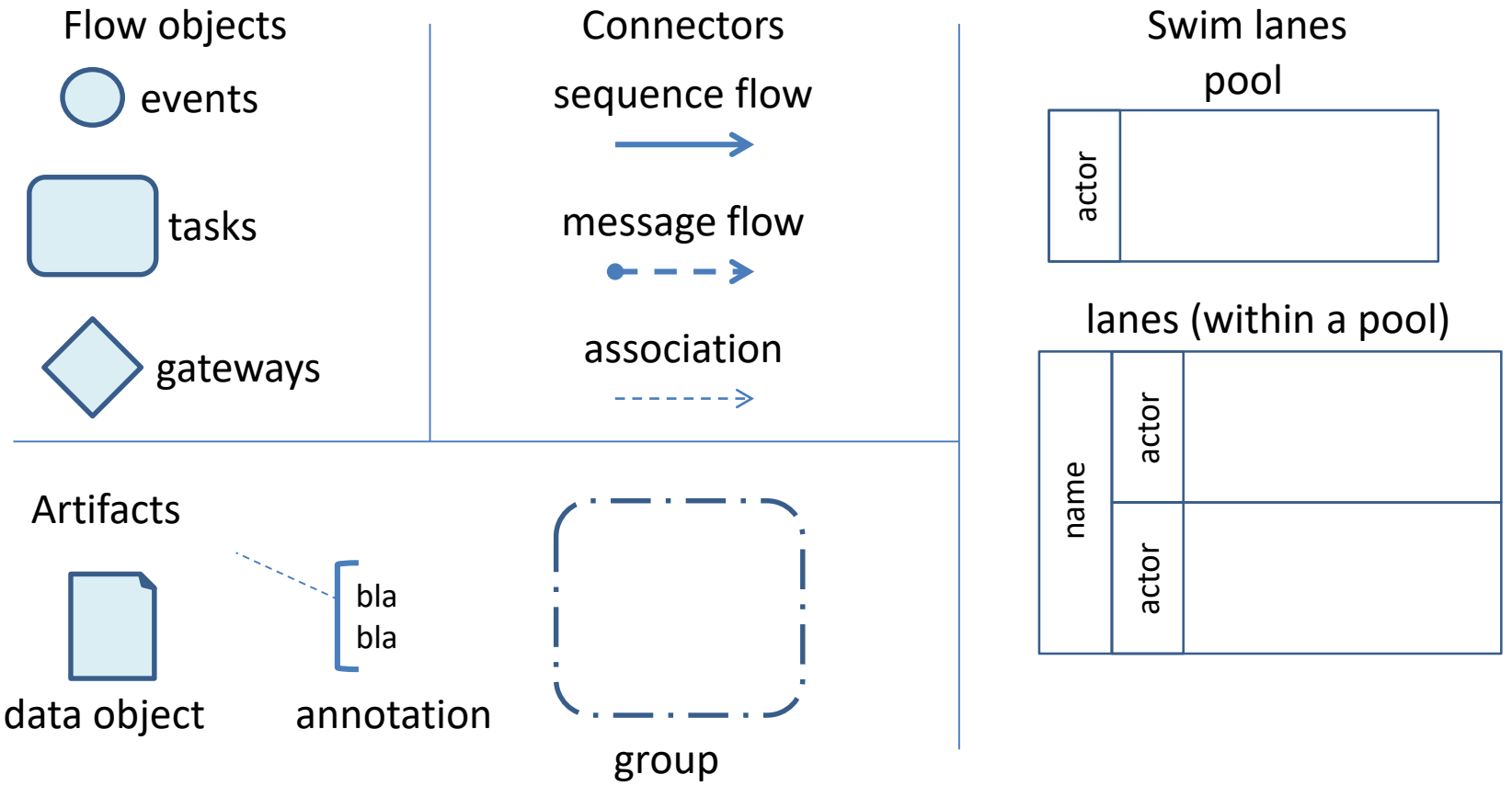
- BPMN models make it possible to describe business process in a **uniform** and **standardized** way, so that all members of an organization can understand each other with a **minimum of ambiguity**.
- BPMN models make it possible to create a bridge between business process and their implementation in IT systems.
- The BPMN formalism is to a large extent independent of any **modeling/analysis methodology** of business processes.
- The BPMN formalism is an **internationally accepted standard** of OMG*.

(*) http://www.bpmn.org/

A BPMN Model for the Loan Demand Assessment



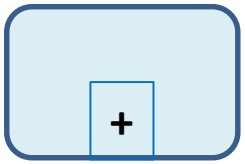
Elements of BPMN Models



Note: the above list is by no means exhaustive. It just shows some general principles of the BPMN formalism as well as its most widely used constructs.

Specialized Tasks

The BPMN formalism provides specialized constructs for certain types of tasks (non exhaustive list):



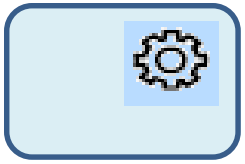
sub-process

send

task repeated a given number of times

reception

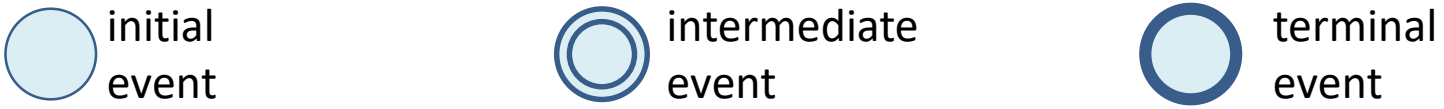
task manually performed by a human operator



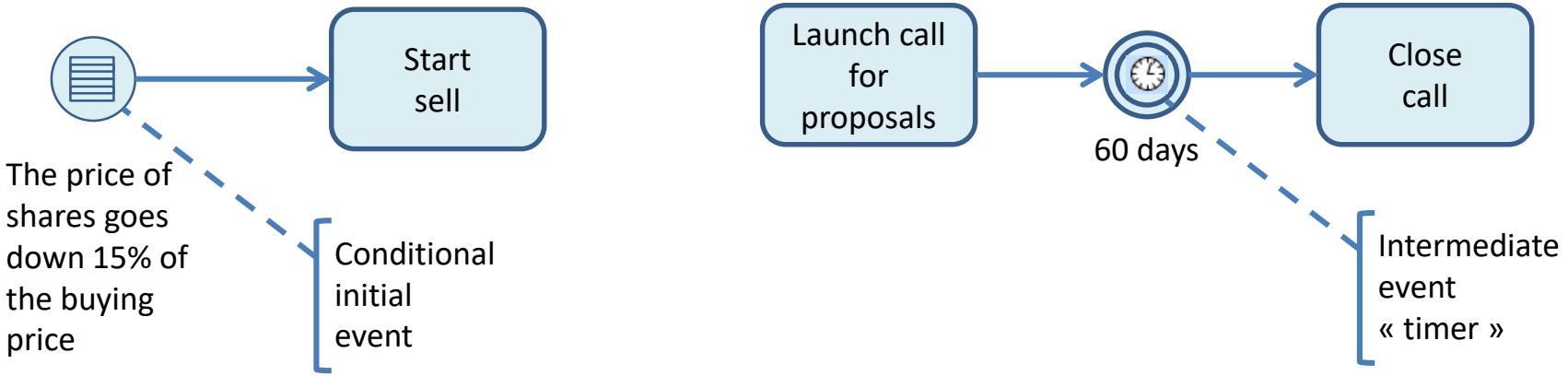
task relying on a external activity or service

Specialized Events

The BPMN formalism distinguishes initial, intermediate and terminal events.

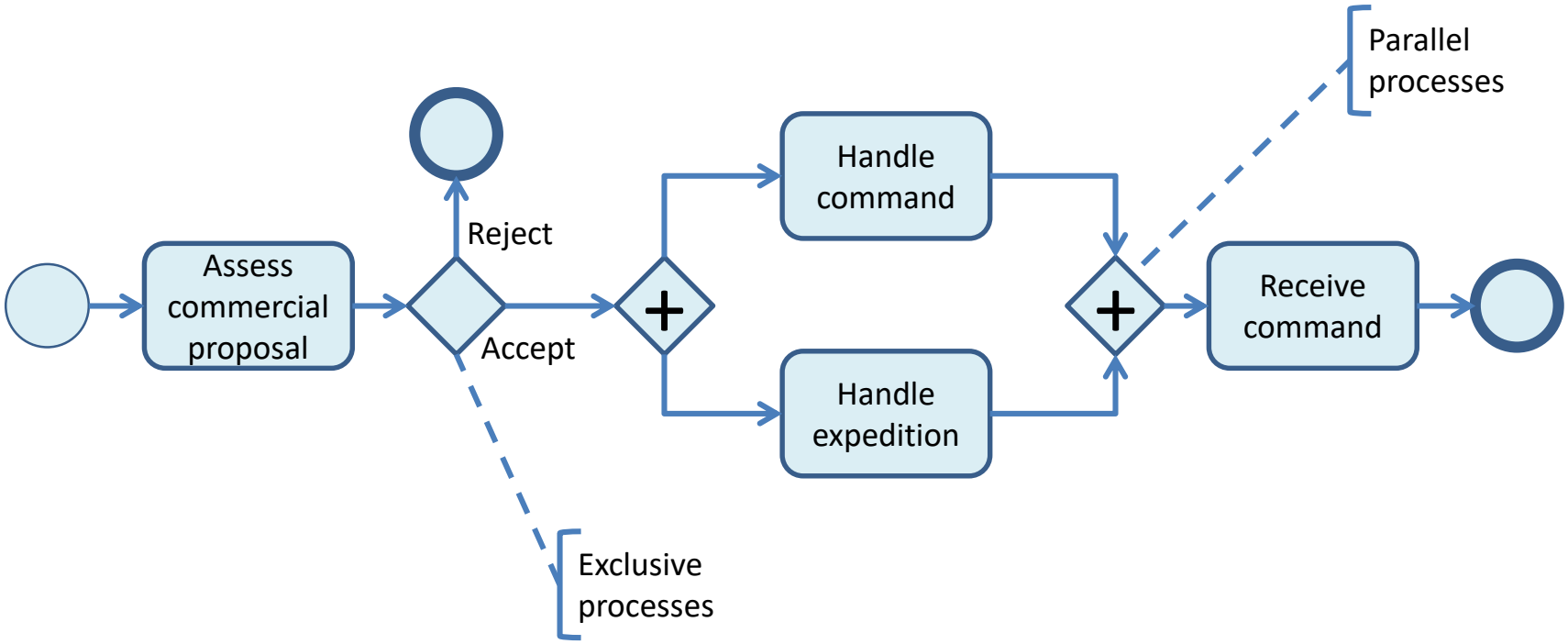


Inside each category, the BPMN formalism provides several specialized events.
For instance:



Specialized Gates

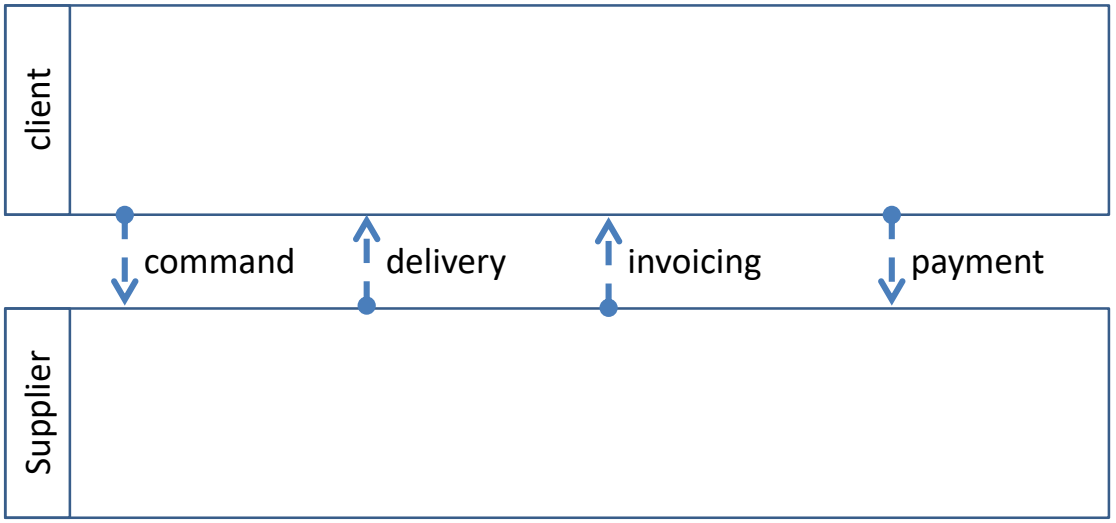
The BPMN formalism provides also several type of specialized gates.
For instance:



Orchestration versus Choreography

From version 2.0, the BPMN formalism distinguishes clearly two types of process coordination:

- **Orchestration** which describes tasks performed by one actor or a group of actors sharing the same data/information (and typically belonging to the same entity). The activities of these different actors are participating to an orchestration are represented in lanes of a swimming pool.
- **Choreography** which implies several separated interacting entities (e.g. client/supplier). It is then assumed that these entities interact by message exchanges but do not share their data/information. Activities are then represented in separated pools.



CentraleSupélec

LECTURE 1. PART 2. SECTION 4. BPMN AS A MODELING LANGUAGE

Model versus Graphical Representation

BPMN models are very convenient to **visualize** business processes. Two important remarks should however be made here:

- When the considered process (or set of processes) is large, it can only be visualized by parts. Therefore, the analyst should **reconstitute** the model in her or his mind.
- The size, color and position of BPMN objects are **meaningless** (and may differ from one tool to another one) although they can be useful for a better visualization. If some computerized operations are performed on the model, e.g. generate web apps from the model, they consider only its structure, not its look-and-feel.

The model should not be confused with its graphical representation.

CentraleSupélec

Making BPMN a Textual Language

The BPMN standard includes a **textual representation** of models, based on **XML** (eXtended Markup Language).

For the purpose of this course, we shall design a simpler textual representation of a small subset of the notation.

The first step of defining an **artificial language** consists in designing a **formal grammar** for this language.

A formal grammar is a set of **production rules** for "phrases" of the language. The rules describe **how to form phrases** from the **language's alphabet** that are valid according to the **language's syntax**.

A grammar does not describe the **meaning** of the phrases or what can be done with them in whatever context, only their form.

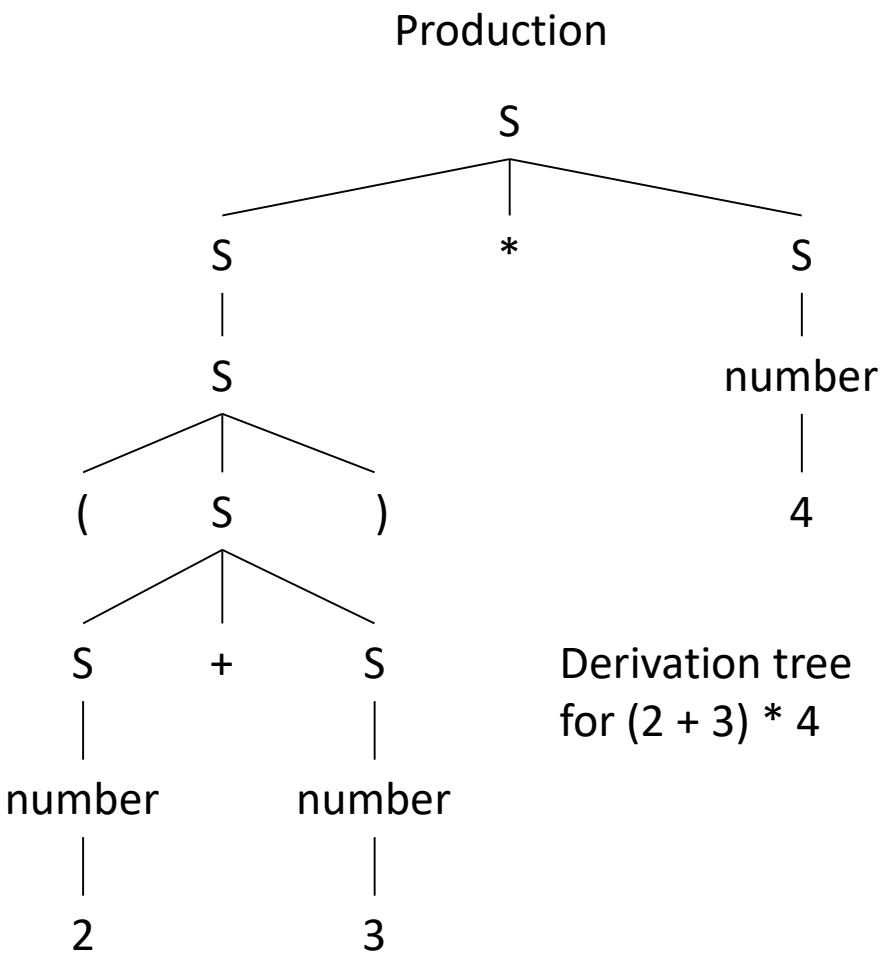
Grammar of Arithmetic Expressions

Alphabet:

- Numbers (1, 5341…)
- Operators “+”, “-”, “*” and “/”
- Parentheses “(“ and “)”

Production rules*:

- S ::= number ;
- S ::= S “+” S ;
- S ::= S “-” S ;
- S ::= S “*” S ;
- S ::= S “/” S ;
- S ::= “(“ S “)” ;
- number ::= any sequence of digits ;



Derivation tree for (2 + 3) * 4

(*) This grammar does not take into account the precedence of operators.

CentraleSupélec

Extended Backus-Naur form

The **Extended Backus-Naur form** (EBNF) is a metasyntax notation that can be used to express a **context-free grammar** (grammars with only one non-terminal symbol in left-hand side of rules). EBNF constructs are the following.

Construct	Meaning	Example
… ::= … ;	definition	S ::= number ;
“…”	terminal symbol	“if”
… …	concatenation	S “+” S
… | …	alternative	S | T
[…]	optional part	“if” Condition “then” instruction [“else” Instruction]
…*	repetition (any number of times)	S*
…+	repetition (at least once)	S+
…?	0 or 1 time	S?
(…)	grouping	(S)

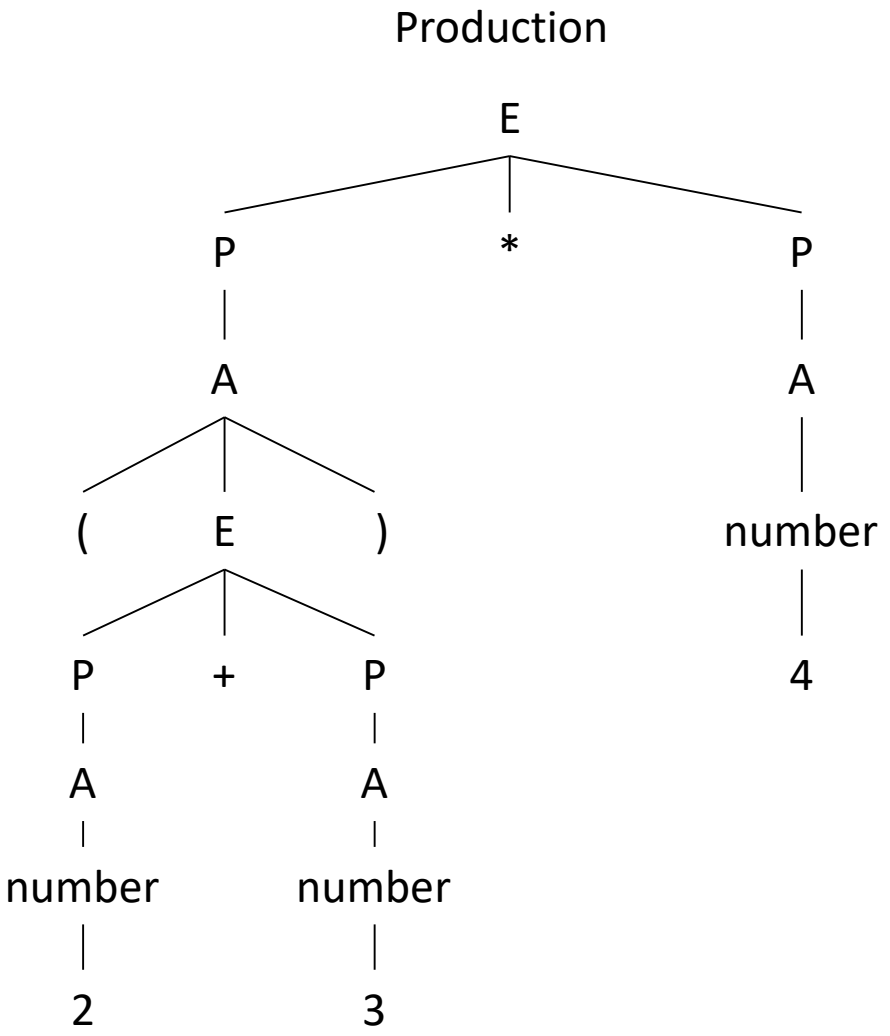
EBNF for Arithmetic Expressions

E ::= P ((“+” | “-”) P)* ;

P ::= A ((“*” | “/”) A)* ;

A ::= number | “(“ E “)“ ;

Derivation tree for (2 + 3) * 4



CentraleSupélec

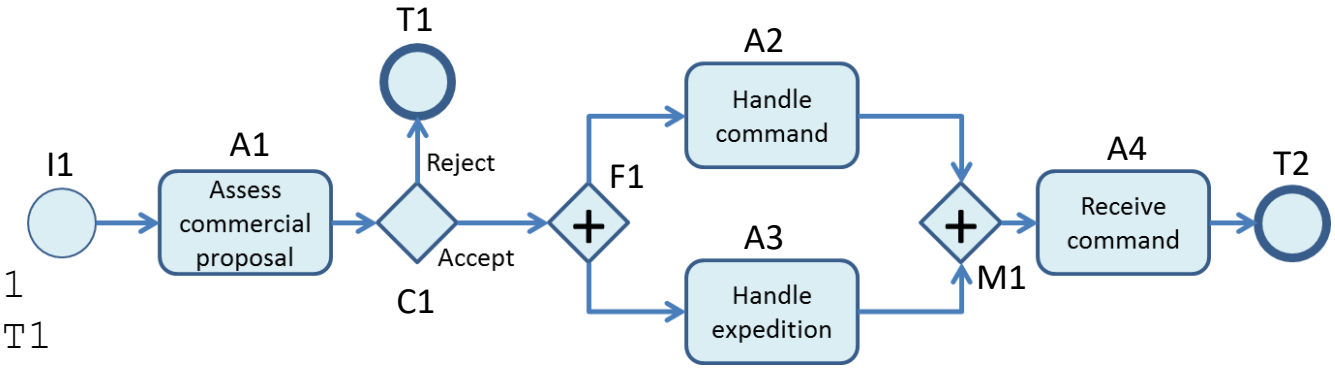
BPMN Elements (Fragment)

Fragment of BPMN that we shall consider:

Type	Objects	Attributes	Ports
Basic Objects	Initial event	identifier	out
	Terminal event	identifier	in
	Activity	identifier, label	in, out
	Choice	identifier	in, out1, out2
	Fork	identifier	in, accept, reject
	Merge	identifier	in1, in2, out
Connectors	Connect		source, target
	Send	message	source, target
Containers	Model	identifier	
	Pool	identifier, label	
	Lane	identifier, label	

CentraleSupélec

Example




```
model Buy
    initial-event I1
    terminal-event T1
    terminal-event T2
    activity A1 "Assess commercial proposal"
    activity A2 "Handle command"
    activity A3 "Handle expedition"
    activity A4 "Receive command"
    choice C1
    fork F1
    merge M1
    connections
        connect I1.out A1.in connect A1.out C1.in
        connect C1.accept F1.in connect C1.reject T1.in
        connect F1.out1 A2.in connect F1.out2 A3.in
        connect A2.out M1.in1 connect A3.out M1.in2
        connect M1.out A4.in connect A4.out T2.in
end
```

CentraleSupélec

EBNF for our BMPN Fragment (1)

Model ::= “model” Identifier ModelObject* ConnectionClause? “end” ;
ModelObject ::= Pool | Lane | FlowObject ;

Pool ::= “pool” Identifier Label PoolObject* ConnectionClause? “end” ;
PoolObject ::= FlowObject | Lane;

Lane ::= “lane” Identifier Label FlowObject* ConnectionClause? “end” ;

FlowObject ::= InitialEvent | TerminalEvent | Activity | Choice | Fork | Merge ;

ConnectionClause ::= connections (Connect | Send)+ ;

Identifier: any sequence of letters and digits starting with a letter.
Label: any sequence of characters surrounded by quotes.

CentraleSupélec

EBNF for our BMPN Fragment (2)

InitialEvent ::= “initial-event” Identifier Label? ;

TerminalEvent ::= “initial-event” Identifier Label? ;

Activity ::= “activity” Identifier Label ;

Choice ::= “choice” Identifier Label? ;

Fork ::= “activity” Identifier Label? ;

Merge ::= “activity” Identifier Label? ;

Connect ::= “connect” Path Path ;

Send ::= “send” String Path Path ;

Path ::= (Identifier “.”)+ “.” Port

Port ::= “in” | “out” | “accept” | “reject”

This grammar is actually a bit too permissive (it allows derivation that are not BPMN models), but it is a good compromise: it is easy to implement and once the derivation tree is built, it is easy to check whether it represents actually a BMPN model.

CentraleSupélec

LECTURE 1. PART 2. SECTION 5. WRAP-UP & ASSIGNMENT

CentraleSupélec

Wrap-Up

- When designing a system, it is of primary importance to design **use cases** to ensure a good understanding of needs.
- Use cases are however too limited to specify fully a system.
- **Business Process Model and Notation (BPMN)** is a powerful way to represent business processes.
- BMPN models can be used:
  - as a **communication tool** amongst stakeholders,
  - under some conditions, to generate automatically applications that implement business processes.
- BPMN are a **graphical**, semi-formal, **notation**, with the associated advantages and drawbacks:
  - easy communication between experts (and to some extent, non-experts),
  - possible ambiguities.
- However, they can be made a **language** with a **formal grammar** (a syntax).

CentraleSupélec

Assignment

Go on the web site of your favorite travel agency (or booking application). Look at the process to book a flight/train/boat ticket (or a hotel).

1. Find out the stakeholders of this business process.
2. Design at least two use cases (write them as shown in these slides).
3. Design a BPMN model for this business process (graphical model).
4. Write the textual representation of this model, according to the formal grammar defined in the lecture.

CentraleSupélec

Recommended Readings

Books or articles on use cases and business process modeling:

Alistair Cockburn. *Writing Effective Use Cases*. Addison Wesley, Boston, MA 02116, USA. ISBN-10: 0201702258, ISBN-13: 978-0201702255. 2000

Sanford Friedenthal, Alan Moore and Rick Steiner. *A Practical Guide to SysML: The Systems Modeling Language.* Morgan Kaufmann. The MK/OMG Press, San Francisco, CA 94104, USA. ISBN-10: 0123852064. ISBN-13: 978-0123852069, 2011

Stephen White and Derek Miers. *BPMN Modeling and Reference Guide: Understanding and Using BPMN*. Future Strategies Inc.. Lighthouse Point, FL, USA. ISBN-10: 0977752720. ISBN-13: 978-0977752720, 2008.

and also:

http://www.bpmn.org/

Books or articles on formal languages:

John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman. Introduction to Automata Theory, Languages, and Computation. Pearson Education International. Upper Saddle River, N.J., USA. ISBN 0321210298. 2003. [Difficult]

CentraleSupélec

Louis Charles Joseph Blériot (1872 -1936) is an airplane designer and one of the pioneer pilot of French aviation. He has been the first to cross the channel on July the 25th onboard of the Blériot XI.
He graduated from Ecole Centrale de Paris

Henri Marie Léonce Fabre (1882 -1984) is a French engineer and pilot. He invented the seaplane in 1910.
He graduated from Supélec.