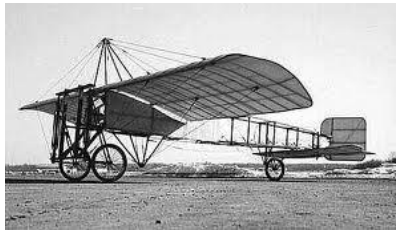
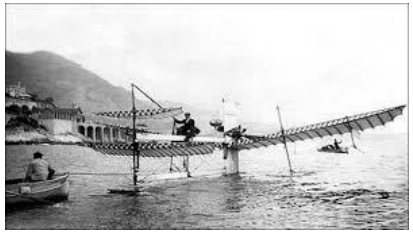
CentraleSupélec

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 2.
SYSTEM ARCHITECTURE & REQUIREMENT ENGINEERING

Notions:

- Operational System Analysis
- Requirement Engineering

CentraleSupélec

Agenda

Part 1. Introduction

Part 2. Use Case: High Integrity Pressure Protection System

Part 3. Architectural Frameworks

Part 4. Operational Analysis

Part 5. Scenarios (Use Cases)

Part 6. Requirements

CentraleSupélec

LECTURE 2. PART 1. INTRODUCTION

Objective of this lecture (1)

Industrial projects became **complex**, especially because of the steadily increasing number of **needs** and **constraints**, sometimes contradictory one another, that should be taken into account. Their management raises important difficulties (delays, cost overrun, bad quality…). The deep reasons for these problems are of systemic essence, i.e. due to the number of **hardware**, **software** and **organizational components** that must be assembled.

System architecture is an emerging discipline which aims at facing these difficulties. More exactly it proposes a **conceptual framework** making it possible:

- On the one hand to **merge** in a coherent way all of the **point of views** on a system, and
- On the other hand, to reason about the system in an accurate way relying on approach by **levels of abstraction**.

CentraleSupélec

Objective of this Lecture (2)

This lecture and the following one are a (too short) introduction to **system architecture**. More exactly they aim at presenting:

- The main concepts of system architecture, in particular the notion of **operational analysis**, **functional architecture** and **physical architecture**.
- The system architecture **process**.
- The **models** (and thus **modeling formalisms**) used to support this process.

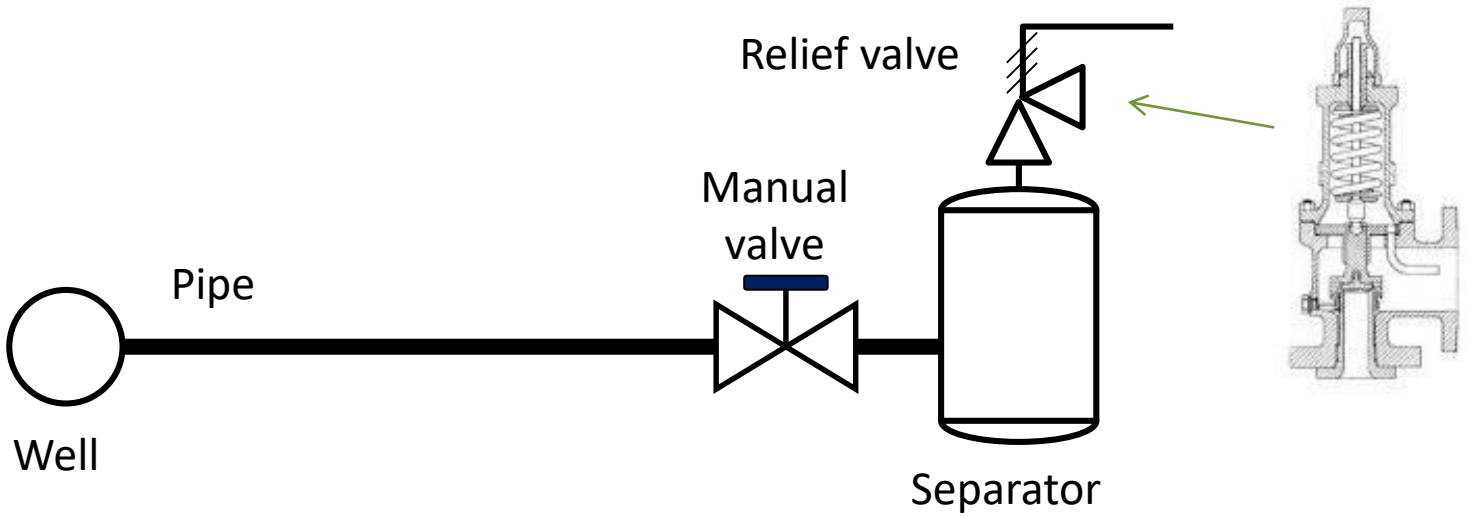
This lecture is deeply inspired by the work of Prof. Daniel Krob (Ecole Polytechnique, France) and the CESAMES method he develops

CentraleSupélec

LECTURE 2. PART 2.
USE CASE: HIGH PRESSURE PROTECTION SYSTEM

High Integrity Pressure Protection System

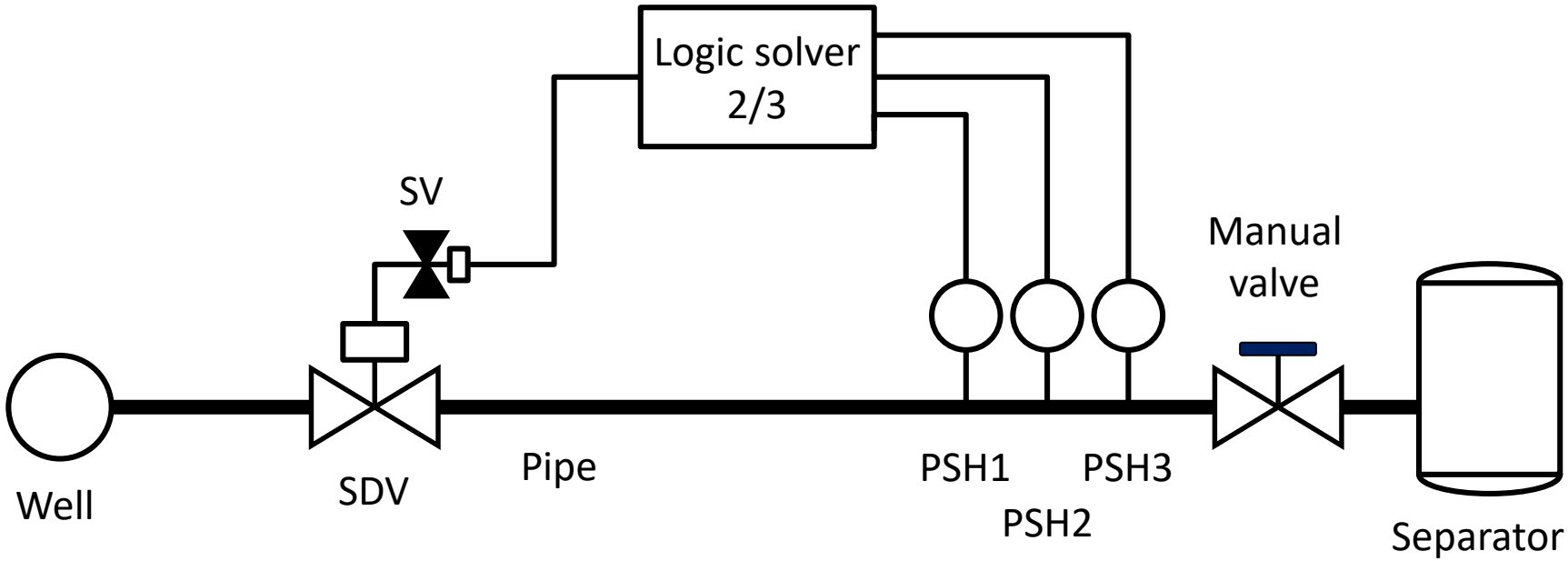
In an oil & gas platform, the mixture of oil, water and gas extracted from the well is sent to a separator through a pipe. To avoid damages to the separator caused by overpressures, a relied valve is installed: when the pressure inside the vessel gets too high, this valve opens and releases the gas which is burnt. Moreover, a manual valve make it possible to stop the inflow, typically to be able to perform maintenance operations.



This protection is relatively cheap and efficient. However, it does not avoid any overpressure and it is not very environment friendly.

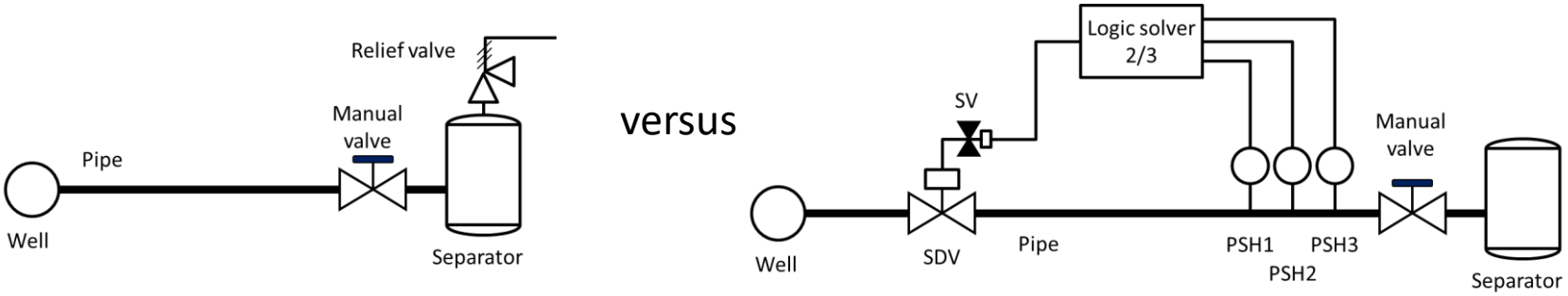
High Integrity Pressure Protection Systems (2)

The question is whether it would be worth to replace the relief valve by a device that prevents overpressures by acting upfront. **High Integrity Pressure Protection Systems** are such devices. Three redundant pressure sensors (PSH1, 2 and 3) detect a possible overpressure and send the information to a 2/3 logic solver. This logic solver activates a solenoid valves SV which in turn close the shutdown valve SDV. This removes eventually the overpressure in the separator



What to do?

Even without speaking about the economic impact of such a change in the installation, we must study carefully its feasibility and its **technical** and **organizational consequences**.



This is typically the purpose of a **system architecture process**.

CentraleSupélec

LECTURE 2. PART 3. ARCHITECTURAL FRAMEWORKS

CentraleSupélec

System Architecture

System Architecture is an emerging discipline that aims primarily at **integrating** other engineering disciplines. Industrial systems are nowadays so complex that the traditional discipline silo-ed, **divide-and-conquer,** approach for system design is no longer sufficient.

System architecture designates both a **process** and the **result of this process**.

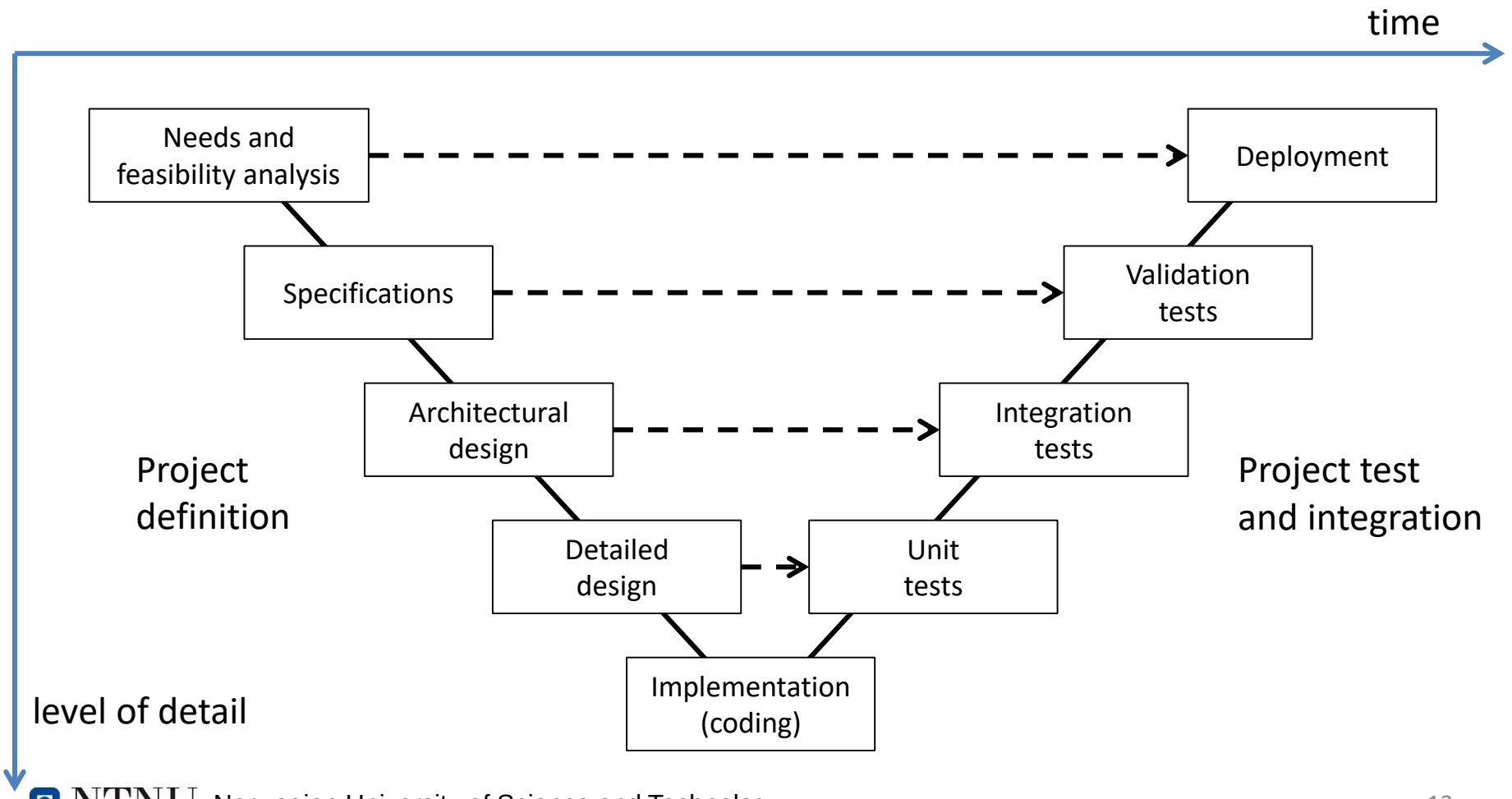
System architects apply methodologies that involve the design of models. These methodologies are often called **architecture frameworks**. The ISO/IEC/IEEE 42010 Conceptual Model of Architecture Description defines the term architecture framework as: "*An architecture framework establishes a common practice for creating, interpreting, analyzing and using architecture descriptions within a particular domain of application or stakeholder community*."

The **SysML** modeling notation is often used to design models, although it has essential drawbacks.

In this course, we shall adopt **Krob Architectural Framework**.

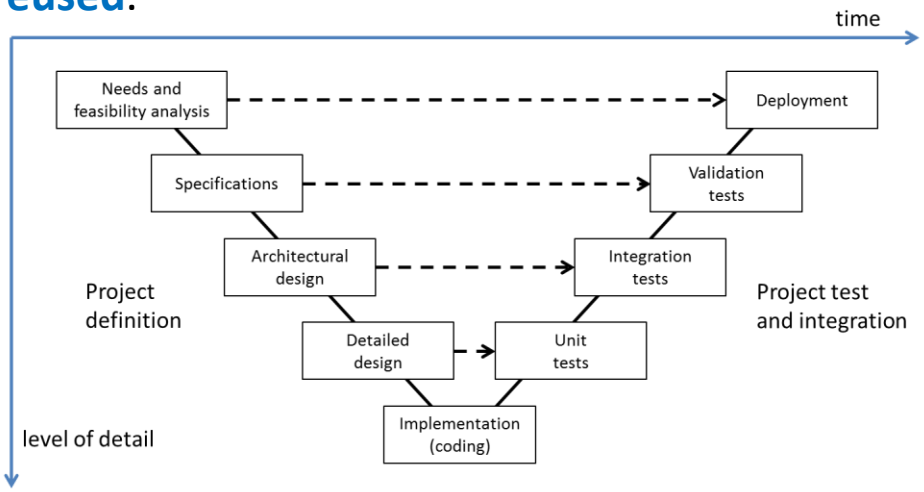
V-Model of System Design

The **V-model** of development, initially designed for software projects, is mentioned in virtually all discourses on system design in general and system architecture in particular.



V-Model of System Design

However, in most of the cases, systems are not designed from scratch. Most of the hardware, software or organizational components of a “new” system are in fact **reused**.



The **V-model** of development is actually more **logical** than **chronological**.

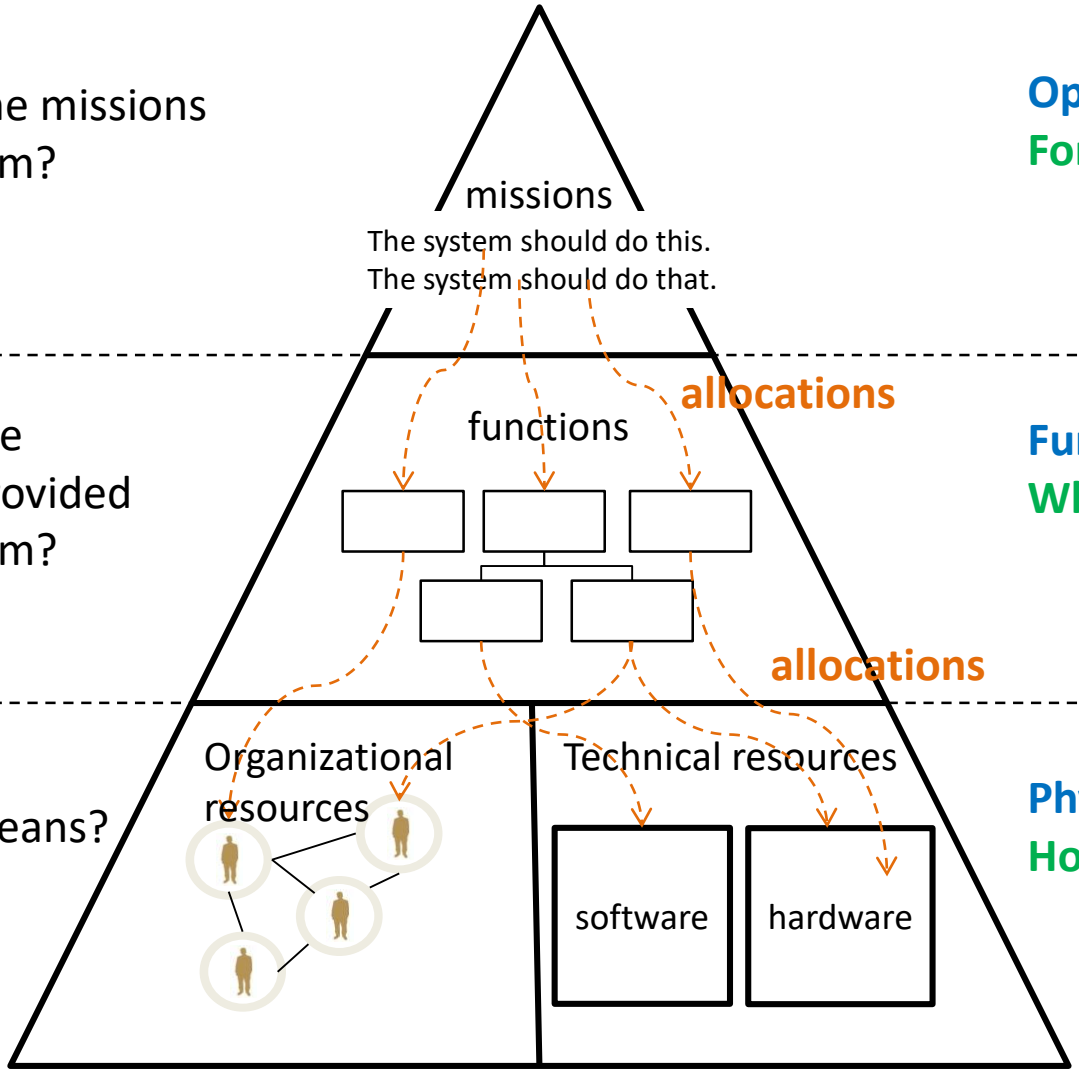
The **system architecture process** aims therefore at **organizing logically** the different **point of views** on the system and to **monitor this organization** through the whole design phase. It is in essence a **concurrent** and **collaborative** process: different point of views are supported by different teams. Moreover, it is an **iterative** process: one cannot expect to find the “good” solution from the very beginning. The “good” solution results always from a process.

Krob Architectural Framework

What are the missions of the system?

What are the functions provided by the system?

By which means?



Operational analysis:
For whom and why?

Functional architecture:
What?

Physical architecture:
How?

CentraleSupélec

Krob Architectural Framework

Points of view	Questions	Analyses	Keywords	Models
Operational	For whom? Why?	Analysis of the environment of the system	Missions, use case, requirement, operational context, life cycle	Interactions of the system with external systems
Functional	What?	Abstract analysis of the system	Function, task, process, mode	Abstract functions of the system
Physical	How?	Concrete analysis of the system	Component, part, architecture, configuration	Concrete components of the system

CentraleSupélec

LECTURE 2. PART 4. OPERATIONAL ANALYSIS

Objectives and Results of the Operational Analysis

The **objective** of the **operational analysis** are:

- To define the **perimeter/boundary** of the system under design.
- To characterize the **environment** of the system, i.e. all **extern systems** with which the system will interact.
- To characterize theses **interactions**.

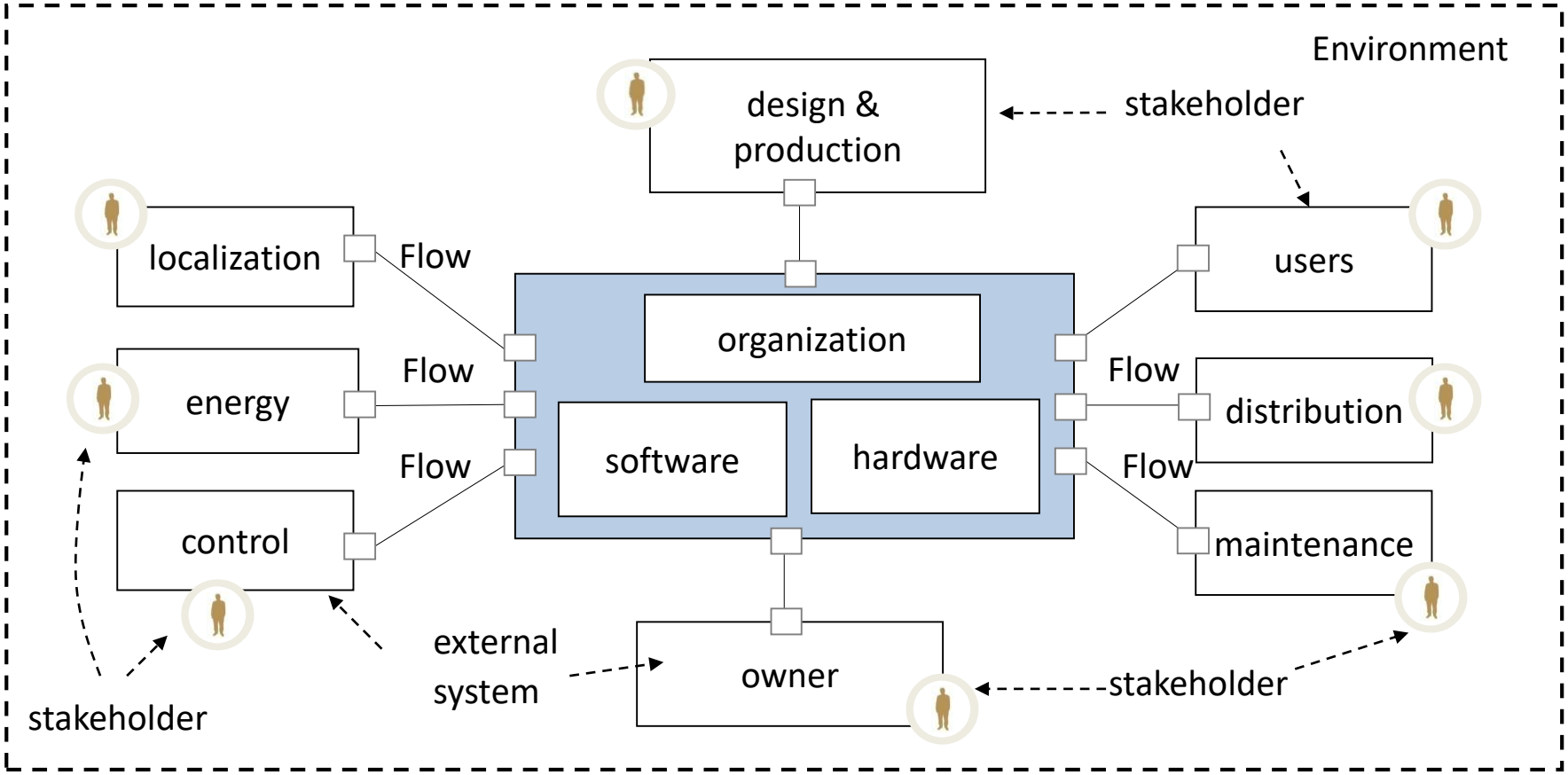
The outcome of the operation analysis is a set of more or less formalized **models** making it possible:

- To define the **perimeter/boundary** of the system and its **environment**.
- To define the **life-cycle** of the system and its different **contexts of operation**.
- To define the **interactions** with its environment in each phase of its life-cycle and/or context of use.

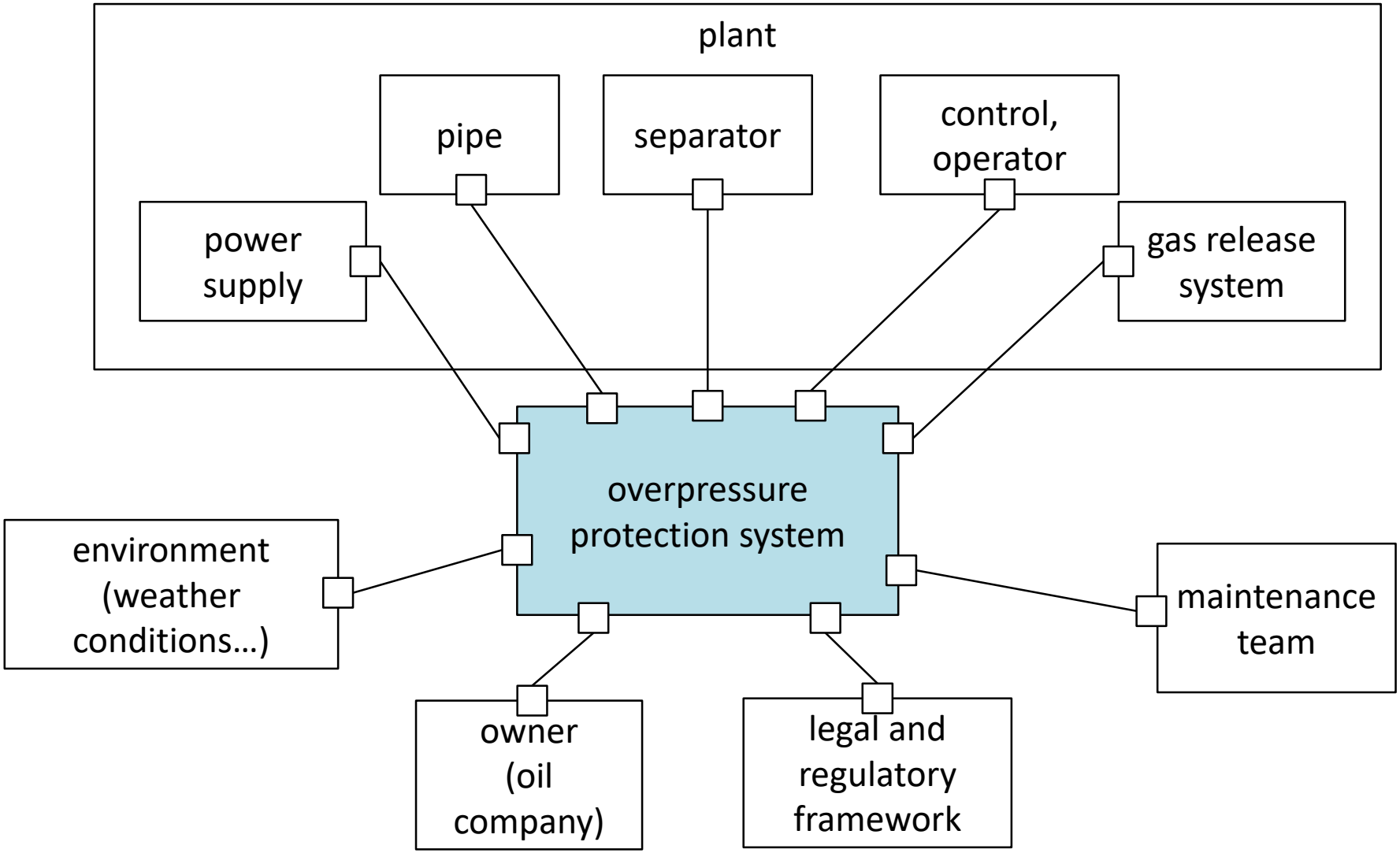
This result is often made contractual as a set of **requirements**. These requirements are referenced, hierarchized and associated with level of maturity.

Environment of System

The **environment** of the system is made of external systems that have an influence on the system. A **stakeholder** is a **human actor** who represents one of the external systems. **Block diagrams** are useful to represent the system and its environment.



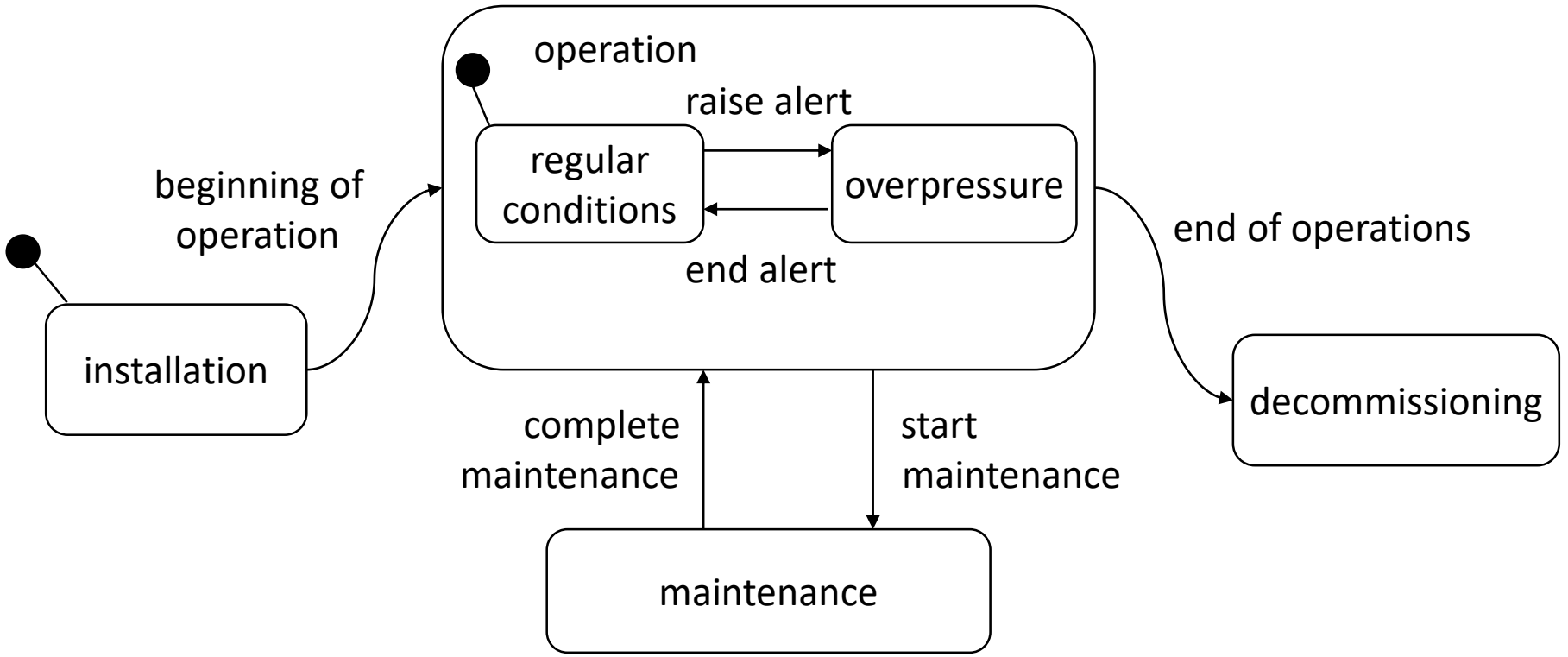
Application to the Use Case



Life-Cycle and Operation Modes

A system goes through different **phases** or **modes** of operations during its **life-cycle**. It changes of mode when some event occurs. Events can be either forecasted/planned or unforeseen. The **missions** of the system differ in general from one mode to another one.

State-charts, also called **state diagrams**, are well suited to represent modes and transitions between modes, i.e. the life-cycle of the system.



CentraleSupélec

Difficulties of Operational Analysis

Difficulty 1: dive into the **solution** before stating the **problem**.

Take the time to state the problem is the only way to ensure that nothing has been forgotten and that all potential solutions have been considered.

Difficulty 2: **forget** some external systems.

Some external systems are either so “obvious” or on the contrary so well “hidden” that it is easy to forget them.

Difficulty 3: **Overload** the analysis of the environment with useless considerations and/no being unable to define correctly the **hierarchy of concerns**.

The objective of the operational analysis is not to study everything in details, but to ensure that we have global and complete vision of the system and of its environment.

Difficulty 4: **Reason too abstractly**.

There is a permanent risk of falling into “**abstract non-sense**”. It is of primary importance to always go back to **concrete issues**, typically by designing **use cases**.

CentraleSupélec

LECTURE 2. PART 5. SCENARIOS (USE CASES)

Operational Scenarios

Use case 1

- Name: Overpressure, version 1
- Description: Alert sequence in case of overpressure
- Actors: Pipe, Separator, Overpressure Protection System, Operator
- References: None
- Prerequisites: The Overpressure Protection System should work correctly
- Consequents: Damages to the Separator avoided.

Sequence of Events

- Due to the injection of high pressure water, the well produces in the Pipe at x m^3/s which induces an overpressure.
- The Overpressure Protection System detects the overpressure.
- The Overpressure Protection System closes the shutdown valve, which isolates the Separator.
- The Overpressure Protection System informs the Operator of the incident.
- The Operator inspects the plant.
- The Operator signals the end of the alert and restarts the production.

Exceptions

- The incident happens during the night. The Operator waits the morning team to restart the production.

We may have forgotten to distinguish “day” and “night” operation phases

Operational Scenarios

Use case number 2

- Name: maintenance of the solenoid valve, version 1;
- Description: part of the maintenance of process of the overpressure protection system;
- Actors: Pipe, Separator, Overpressure Protection System, Operator, Maintenance Team
- References: Use case number 1;
- Prerequisites: None;
- Consequents: None;

Sequence of Events

- The Operator stops the production and isolates it by closing the manual valve.
- The Maintenance Team stops the Overpressure Protection System.
- The Maintenance Team works on the solenoid valve.
- The Maintenance Team tests the solenoid valve.
- The Maintenance Team restarts the Overpressure Protection System.
- The Operator restarts the production.

Exceptions

- None.

Do we still need a manual valve?

Incident/Accident Scenarios

Use Case number 3

- Name: accident due to overpressure, version 1;
- Description: event sequences leading to an accident due to an overpressure;
- Actors: Pipe, Separator, Overpressure Protection System
- References: Use Case number 1;
- Prerequisite: malfunction of the solenoid valve;
- Consequents: damage or blow-up of the separator;

Sequence of Events

- Due to the injection of high pressure water, the well produces in the Pipe at x m^3/s which induces an overpressure.
- At least two out of the three sensors detect the overpressure.
- The calculator orders the solenoid valve to close the shutdown valve.
- Because of the malfunction the solenoid valve cannot close the shutdown valve.
- The overpressure leads to a rupture of the separator confinement.

Exceptions

- None.

Should we install an alarm that warn the operator to stop the production to prevent too much damages?
Is the probability of this scenario low enough?

Incident/Accident Scenarios

Use Case number 4

- Name: Spurious shutdown, version 1 ;
- Description: Spurious shutdown of the production due to a malfunction of sensors;
- Actors: Pipe, Separator, Overpressure Protection System
- References: Use Case number 1 ;
- Prerequisite: malfunction of sensors PSH1 and PSH2 ;
- Consequents: loss of production ;

Sequence of Events

- The well produces in the pipe x m^3/s which is in acceptable limits.
- The sensor PSH1 is failed (detected).
- The sensor PSH2 sends an erroneous overpressure alert.
- The calculator orders the solenoid valve to close the shutdown valve.
- The solenoid valve closes the shutdown valve.
- The production is stopped until the operator checks everything

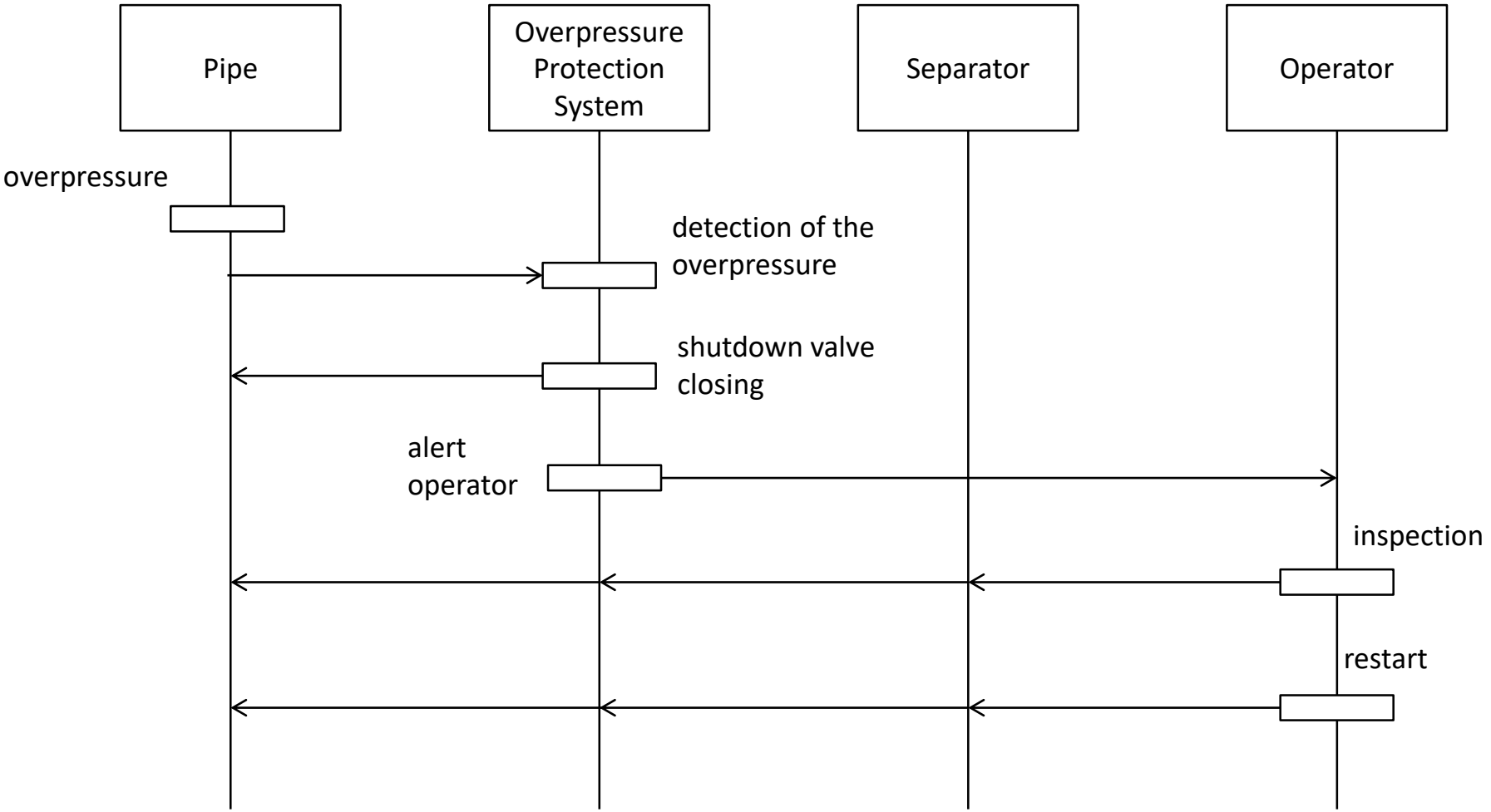
Exceptions

- None.

Should we change the control logic?
Is the probability of this scenario low enough?

Models of Operational Scenarios

Sequence Diagrams are well suited to visualize operational scenarios.



Interactions

Interactions between the system and external systems must be characterized

External System	Phase/Mode	Interaction
Electric Power Supply	• operation • maintenance	The Electric Power Supply supplies power to the sensors and the calculator.
Electric Power Supply	• operation / overpressure • maintenance	The Electric Power Supply supplies power to the solenoid valve.
Pipe	• operation	The Overpressure Protection System detects overpressures in the Pipe.
Pipe & Separator	• operation	The Overpressure Protection System isolates automatically the Separator in case of an overpressure in the Pipe.
…	…	…
Laws & Regulations	all	The regulation sets the maximum number of m^3 of gas per year the plant is allowed to reject in the atmosphere.

CentraleSupélec

LECTURE 2. PART 6. REQUIREMENTS

Requirements

The (implicit or explicit) outcomes of the operational analysis are in general **formalized** as corpuses of **requirements**.
According the IEEE Standard 1220-1998 of systems engineering:
A **requirement** R(S) on a system S is a **non ambiguous, testable and measurable property** that expresses a characteristic of a constraint that the system **should satisfy** to be **accepted** by the **stakeholders**.

Name	**Reference**	**Maturity**
<<name>>	<<reference>>	e.g. weak/average/strong
Statement		
The <<system>> should <<do something>> with <<this level of performance>> in <<that context>>		
Justification		
Justification of the requirement (e.g. demand of a stakeholder)		
Satisfaction criterion		
Means to test that the requirement is fulfilled by the system.		

Categories of Requirements

Requirements are often defined as a property of the system that answers the **need** of an external system.
But then what is a need?
Eventually, a need is a requirement of the external system.
Needs and requirements are intertwined notions and are partly dual.

It is common to distinguish between **functional requirements** and **non-functional requirements**.

- Functional requirements correspond to what the system should be or should do.
- Non-functional requirements correspond to the expected performance of the system, e.g. in terms of weight, cost of ownership, answer delay, safety, reliability….

This distinction is for a good part arbitrary.

CentraleSupélec

Requirements and Contracts

A system is designed to answer needs of external systems. But to be operated in expected conditions and with the expected performance, it has itself needs. These needs are requirements for (some) external systems.

For instance, the High Integrity Pressure Protection System aims at protecting the separator from overpressures. It is designed only for certain range of overpressure (performance). But it needs a power supply (need).

Requirements are more and more often used to support **contractual relationships** between **ordering organizations** and **suppliers**. They specify the expected service in both directions.

To know how to **write a corpus of requirements** is part of the **expected skills** of engineers.

Application to the Use Case

Name	**Reference**	**Maturity**
Overpressure detection	EXG001	high
Statement		
The system should detect a pressure larger than xxx g/cm^2 when in operation.		
Justification		
This xxx corresponds to 80% of the maximum acceptable pressure in the separator.		
Satisfaction criterion		
Test of the sensors in simulated environment		

Application to the Use Case

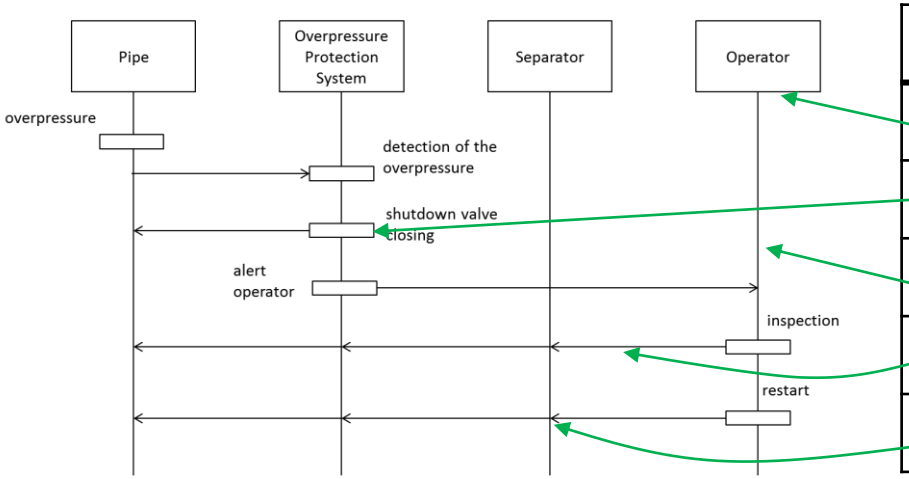
Name	**Reference**	**Maturity**
Availability	EXG033	high
Statement		
The system should be available at least 99% of the time		
Justification		
Planned maintenance operations and failures should not stop the production more than 3 days a year.		
Satisfaction criterion		
RAMS analysis		

CentraleSupélec

LECTURE 2. PART 7.
SYSTEM ARCHITECTURE LANGUAGES

Making Sequence Diagrams a Textual Language

As for BPMN in the previous lecture, there is clearly a need for a textual representation of other architecture models such as sequence diagrams. Here follows the elements of these models (this list may be non exhaustive):



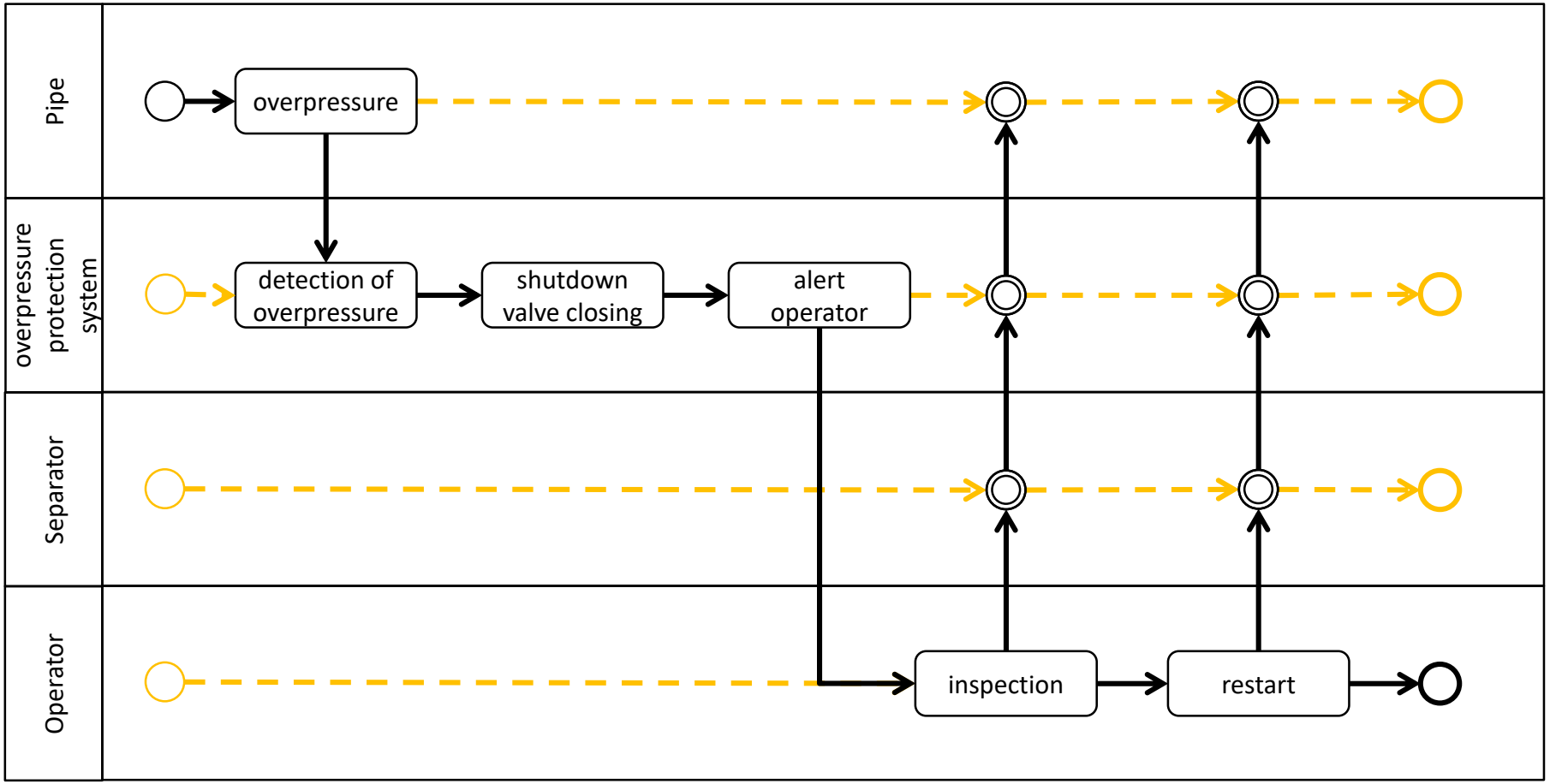
Modeling Elements	What they represent
Lanes	What actors are doing
Actions	Actions of the actors
Top-down connections	Time elapse
Transversal connections	Dependencies between actions
Crossings	Synchronization points

Textual Format

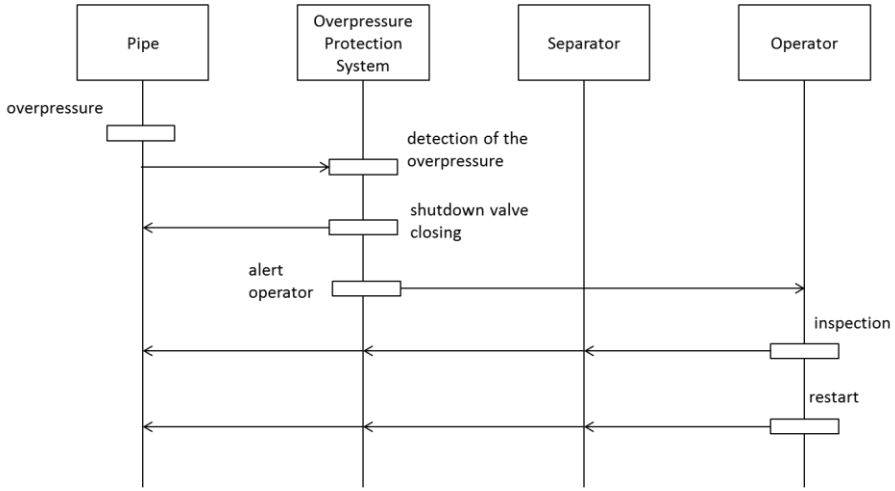
```
model UseCase1
  lane Pipe
    initial-event I
    activity A "Overpressure"
    connections
      connect I.out A.in
  end
  lane OverpressureProtectionSystem
    activity D "Detect overpressure"
    …
  end
  lane Separator
    …
  end
  lane Operator
    …
  end
  connections
    connect Pipe.A.out OverpressureProtectionSystem.D.in
    …
end
```

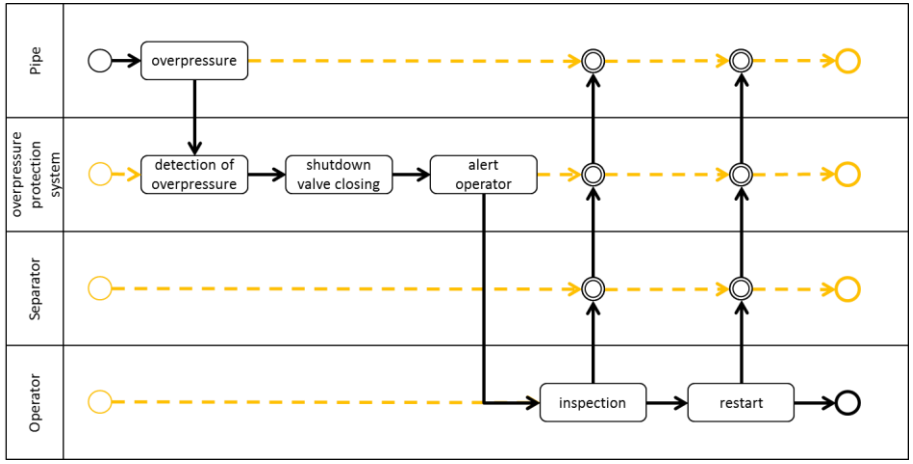
Does it remind you something?

An Equivalent Sequence Diagram



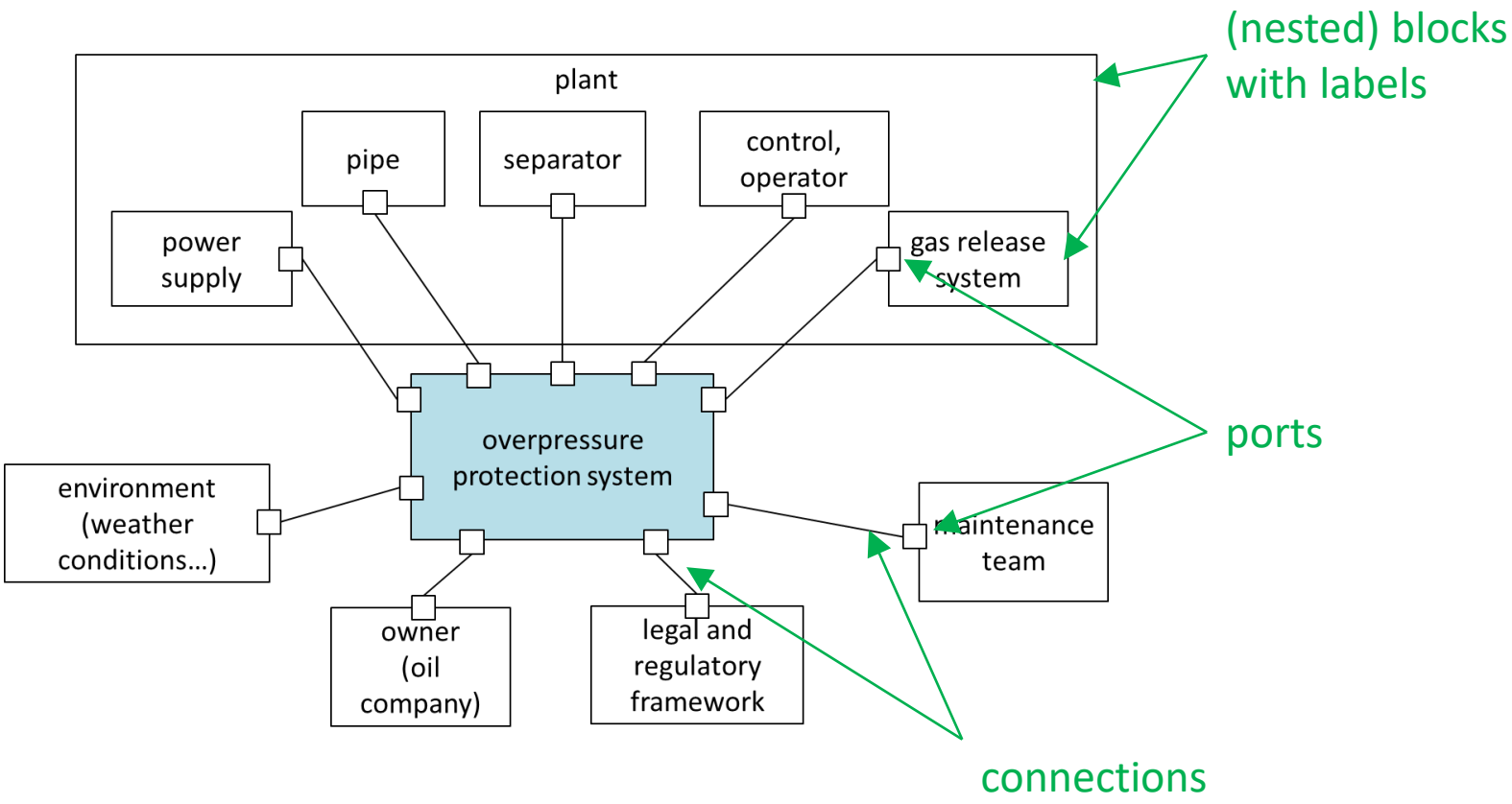
BPMN versus Sequence Diagrams





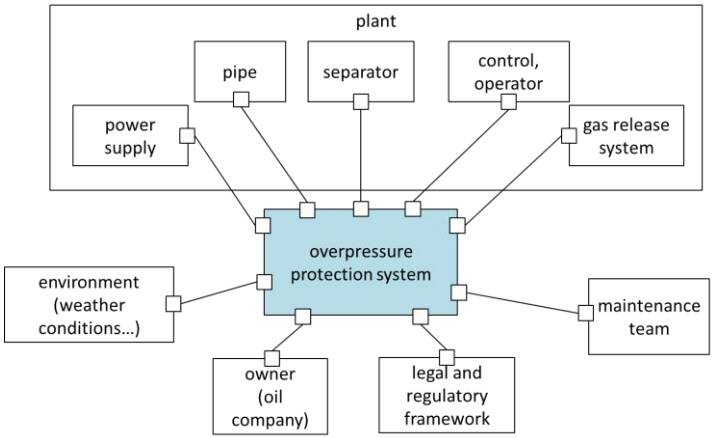
Not all of the sequence diagrams can be casted into BPMN models (the reverse is obviously false as well). However, this example shows that it is of importance to always consider **mathematical/algebraic structures** of models, and not to be stuck on their graphical representations.

Textual Format for Environment Diagrams



Textual Format for Environment Diagrams

```
model OverpressureProtectionSystemEnvironment
  block Plant
    Block Pipe
      port P "connection to system"
      …
    end
    …
  end
  block OverpressureProtectionSystem
    …
    port P3 "connection to pipe"
    …
  end
  …
  connections
    connect Plant.Pipe.P OverpressureProtectionSystem.P3
    …
end
```



CentraleSupélec

EBNF for Environment Diagrams

Model ::= “model” Identifier Label? ModelBody “end” ;
ModelBody ::= Block* ConnectionClause?

Block ::= “block” Identifier Label? BlockBody “end” ;
BlockBody ::= (Block | Port)* ConnectionClause?

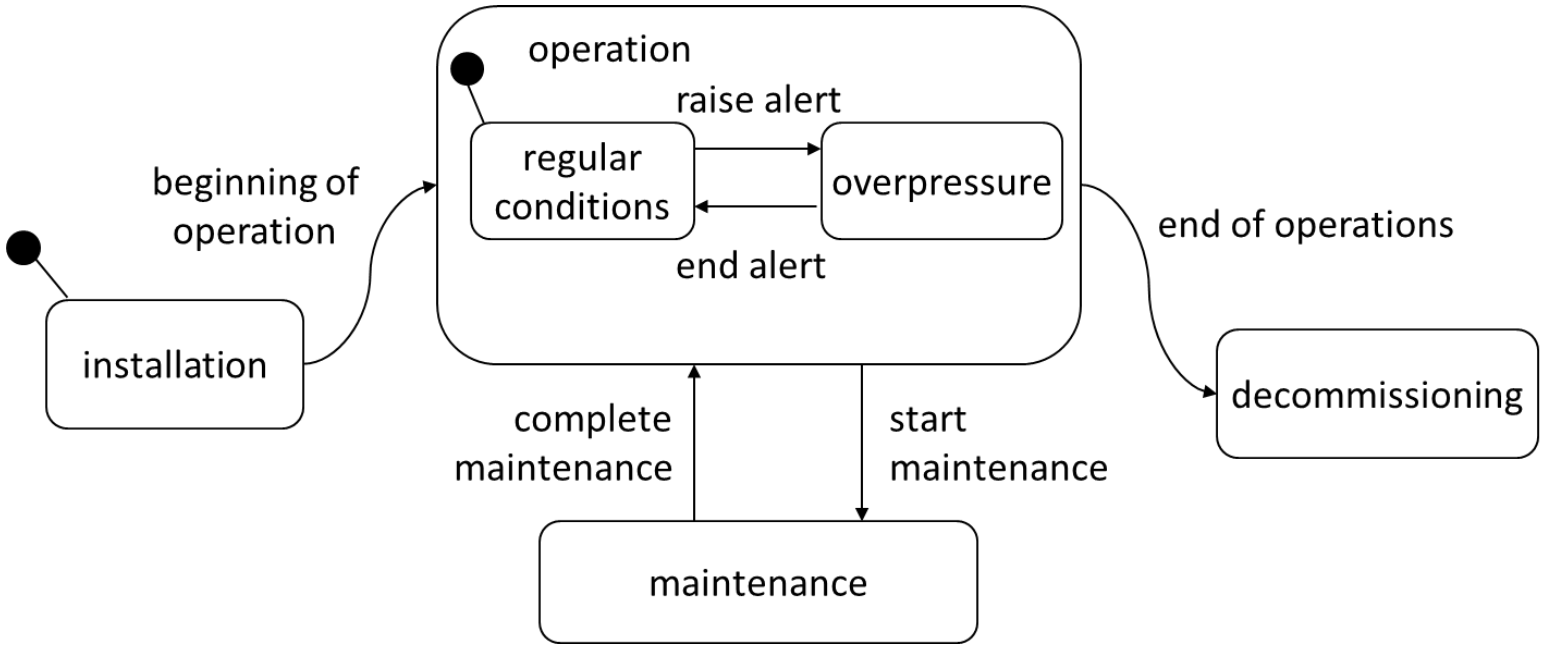
Port ::= “port” Identifier Label?

ConnectionClause ::= “connections” Connection+
Connection ::= “connect” Path Path
Path ::= Identifier (“.” Identifier)*

Identifier ::= any sequence of letters, digits and underscores starting with a letter or an underscore
Label ::= any sequence of characters surrounded with quotes

Textual Format for State Diagrams

The textual format for state diagrams will be discussed on the lecture on automata.



CentraleSupélec

LECTURE 2. PART 8.
WRAP-UP AND ASSIGNMENT

CentraleSupélec

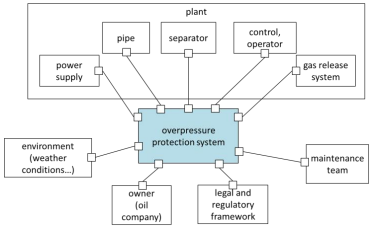
Wrap-Up

- **System Architecture** is an emerging discipline that aims primarily at **integrating** other engineering disciplines.
- System architects apply methodologies that involve the design of models. These methodologies are often called **architecture frameworks**.
- The **Operational Analysis** is logically (but not necessarily chronologically) the first phase of the system architecture process.
- The **Operational Analysis** aims at answering the questions "**why** the system is designed?" and "**for whom** it is designed". It aims defining precisely the boundary of the system as well as at characterizing the **life-cycle** of the system and its **interactions** with **extern systems**.
- The outcomes of the operational analysis are more and more often formalized as a **corpus of requirements**.
- A **requirement** on a system is a **non ambiguous, testable and measurable property** that expresses a characteristic of a constraint that the system **should satisfy** to be **accepted** by the **stakeholders**.
- This corpus is used to establish the **contractual relationships** between the organization which orders the system and its suppliers.

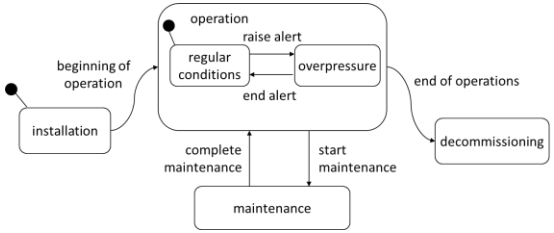
Modeling Formalisms

In this lecture, we used different types of diagrams. They are all more or less parts of the **SysML** graphical notation.

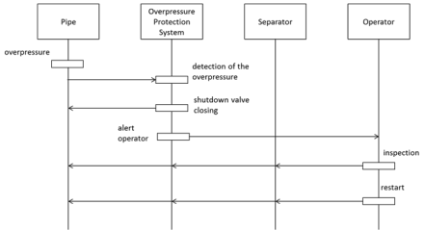
Environment Diagrams



State Diagrams

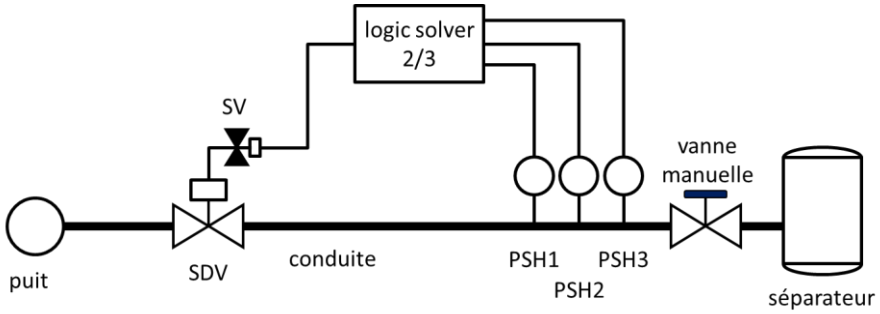


Sequence Diagrams



- All these models aim at easing the **communication** between the stakeholders. Therefore, they must be clear, simple and cannot pretend to be exhaustive.

- With that respect, never forget that Process & Instrumentation Diagrams are a very good communication mean (unfortunately not available in SysML).



Assignment (1)

You are an expert from a company in charge of installing and maintaining elevators in buildings open to the public.

You are called by a nursing home (that hosts senior people) to make a proposal to modernize their elevators.

You have thus to study carefully the client's needs.

Assignment (2)

Make a full operational analysis for the elevator system. This includes:

1. Determining the stakeholders of the elevator system.
2. Characterizing interactions of the elevator system with its external systems.
3. Determining the life-cycle of the elevator system.
4. Designing operational uses cases as well as incident/accidental use cases.
5. Designing a corpus of requirements.
6. Designing models for all the above items (both in graphical and textual format).
7. …

CentraleSupélec

Recommend Readings

Books or articles about System Architecture:

Benjamin S. Blanchard and Wolter J. Fabrycky. Systems Engineering and Analysis. Pearson. Upper Saddle River, NJ 07456, USA. ISBN 978-0137148431. 2008.

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 978-0123852069. 2011.

INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, fourth edition. David D. Walden, Garry J. Roedler, Kevin J. Forsberg, R. Douglas Hamelin and Thomas M. Shortell Ed.. Wiley-Blackwell. Hoboken, NJ, USA. ISBN 978-1118999400. August, 2015.

Daniel Krob website: http://krob.cesames.net/

Books or articles about Requirement Engineering:

Klaus Pohl and Chris Rupp. Requirements Engineering Fundamentals. O'Reilly. Sebastopol, California, USA. ISBN 978-1933952819. May, 2011.

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.