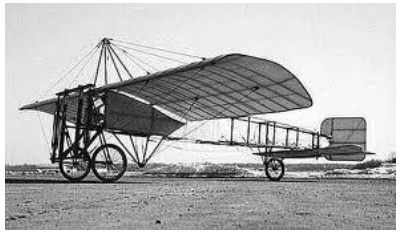
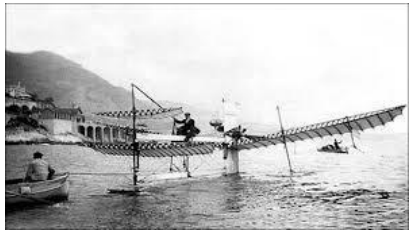
CentraleSupélec

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 1
GENERAL INTRODUCTION

Notions:

- Systems
- Models of Systems
- Modeling Formalisms

CentraleSupélec

LECTURE 1. PART 1. SECTION 1. THE SCIENCE OF MODELS

What does “system” mean?

- We call **system** “a combination of interacting elements that are organized so to achieve one or several established objectives” (INCOSE(1) 2007).
- This course deals more specifically with **intentionally designed technical and sociotechnical systems**, i.e. de systems designed by means of an engineering process:
  - Industrial **products** ;
  - **Organizations** that design, produce, operate and/or regulate these products;
  - **Processes** by which these organizations design, produce, operate and/or regulate these products;
- We shall not consider:
  - Natural systems (molecules, cells, volcanos, rivers…)
  - Human systems not resulting of an engineering process (cities, internet…)
  - Software

(1) International Council on Systems Engineering, http://www.incose.org/

CentraleSupélec

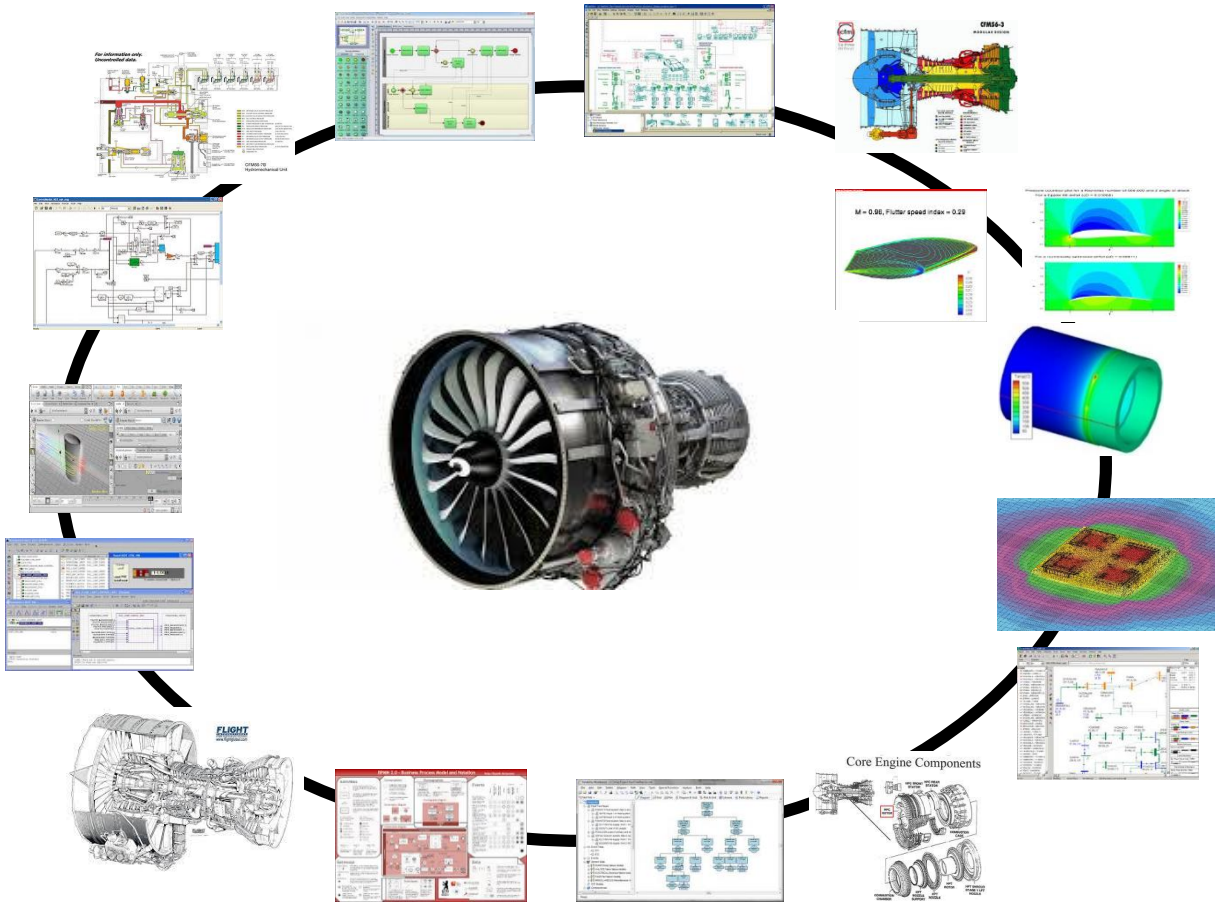
Models are everywhere

- The **systems** designed by industry are more and more **complex**. Not only these **products** are more and more complex but also the **processes** by which they are **designed/produced/operated/decommissioned** and **organizations** that implement these processes are.
- To face this complexity, the different engineering disciplines (mechanics, thermic, electric and electronic, software, architecture…) virtualize their contents to a large extent, i.e. they are designing **models**. We entered the era of:

Model-Based Systems Engineering

- Each system comes with dozens of models. More and more of these models are **embedded** into systems and used for their operation.

Systems in Silico



CentraleSupélec

Science and engineering of models

Models must be taken seriously and considered as **first class citizens**. This raises a number of challenges:

- Better understand the **nature** of models and their **roles** in industrial processes.
- Develop the "**Art of Modeling**"(*) in each and every engineering discipline.
- **Manage** models throughout the **life-cycle** of systems.
- Design tools and methods to support the **integration** of engineering disciplines/processes through the integration of models they produce.
- …

The emerging science of complex systems
is the science of models

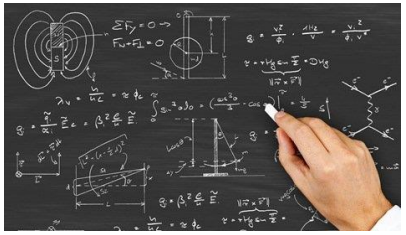
(*) In reference to Knuth's famous series of books about "The Art of Programming"

CentraleSupélec

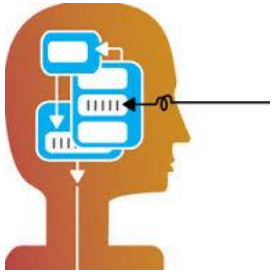
LECTURE 1. PART 1. SECTION 2. WHAT IS A MODEL?

What is a model?

Star

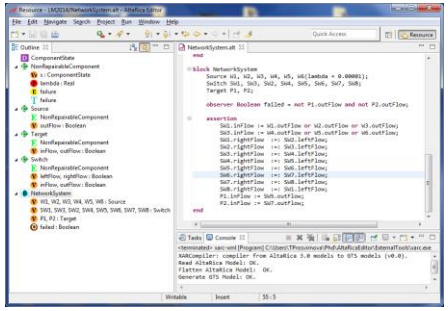


Mathematical Model

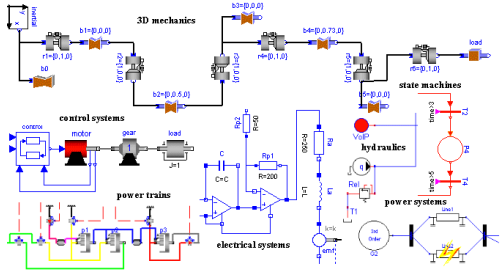


Cognitive Model

Mockup



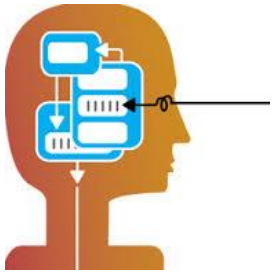
Code



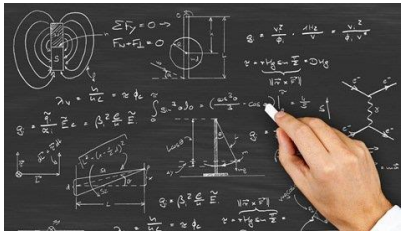
Graphical Representation

All these “things” are models in some way

Models in systems engineering



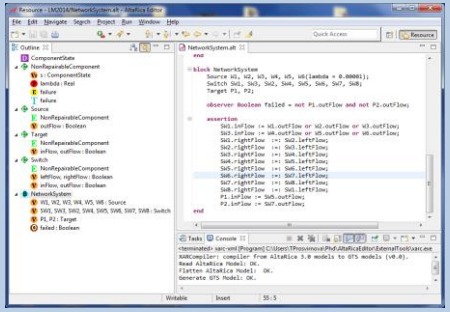
Cognitive Model



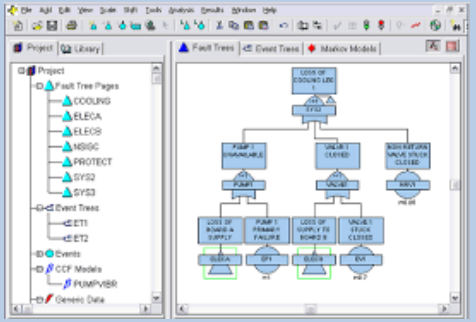
Mathematical Model

mind & paper models

computerized models



Code

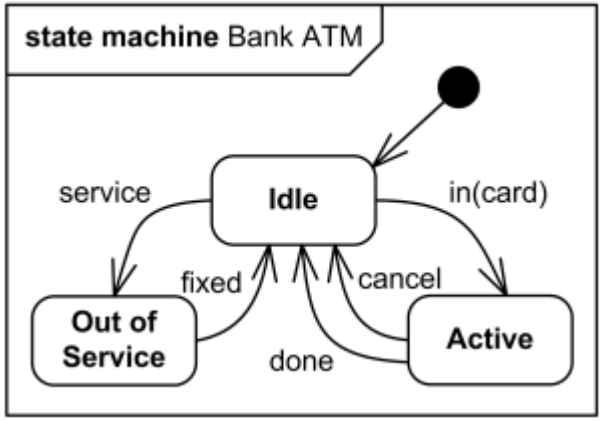


Graphical Representation(s)

Models are **working tools**, not (platonic) ideals the system should comply to.

Modeling languages

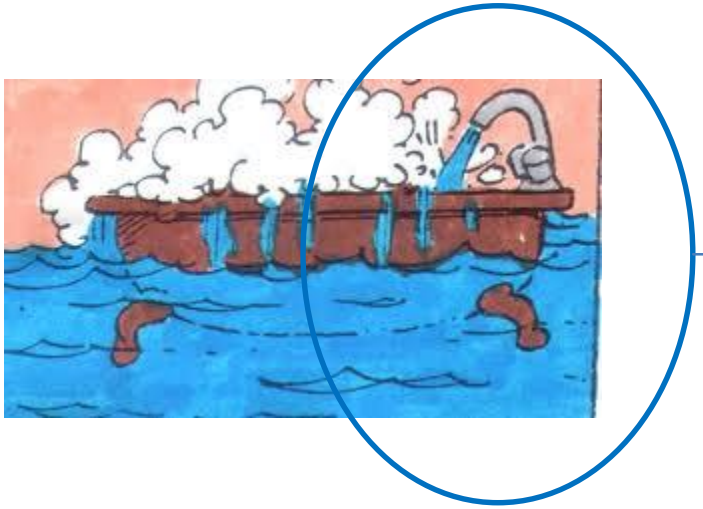
A computerized model is eventually nothing but a sequence of symbols that obeys certain rules. To design a model, we need a **modeling language** (would it be purely graphical), just as to design a program, we need a programming language.
Purely graphical languages reach quickly their limits when models get complex. Textual modeling languages are then necessary. Parts of a large model can be represented graphically, but they should not be confused with the model itself.




```
domain State {Idle, Active, OutOfService}

automaton BankATM
    State state(init = Idle);
    event service, fixed, in(card), cancel, done;
    transition
        service: state==Idle -> state:=OutOfService;
        fixed: state==OutOfService -> state:=Idle;
        in(card): state==Idle -> state:=Active;
        cancel: state==Active -> state:=Idle;
        done: state==Active -> state:=Idle;
end
```

Specific purposes, specific models

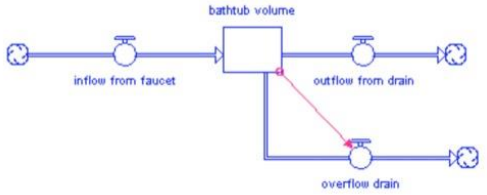


The **content** and the **level of abstraction** of a model depends on what is to be observed, i.e. on the **virtual experiments** to be performed on that model.

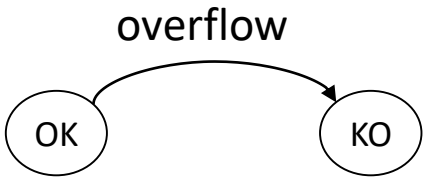
Fluid mechanics

$$\frac{\partial \vec{v}}{\partial t} + \left(\vec{v} \cdot \vec{\nabla}\right) \vec{v} = -\frac{1}{\rho}\vec{\nabla} p + \nu \nabla^2 \vec{v} + \vec{f}$$

Multiphysics simulation



Safety analyses

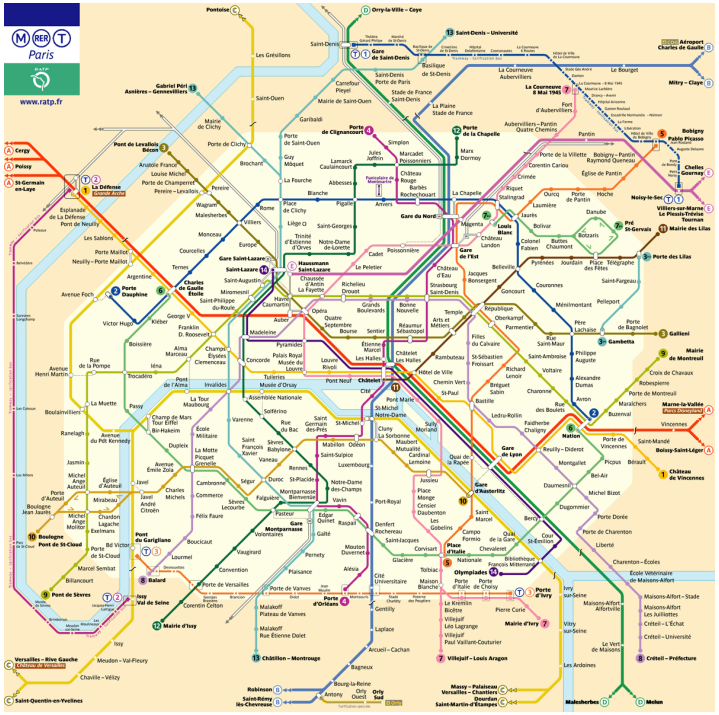


Insurance

Region	Premium price (Winter 2013/14)	Premium price (Winter 2014/15)	Percentage decrease year on year
London North West (NW)	£182.04	£149.05	-18.1%
Hereford (HR)	£113.42	£97.57	-14%
London West (W)	£157.24	£135.68	-13.7%
Enfield (EN)	£154.31	£133.61	-13.4%
Manchester (M)	£137.45	£121.22	-11.8%
Cambridge (CB)	£118.16	£104.45	-11.6%
Liverpool (L)	£138.86	£123.41	-11.1%
Southend-on-sea (SS)	£150.17	£133.64	-11%
Harrogate (HG)	£122.63	£109.32	-10.9%
Huddersfield (HD)	£128.56	£114.65	-10.8%

All models are false. Some are useful.

George Edward Pelham Box
(1919-2013)



A model is an **abstraction** and a **point of view** on the system at hand.
It is designed with a particular **purpose**, i.e. to realize particular **virtual experiments**, to **learn lessons** from these experiments and to **make decision** about the system

Taxonomy of models

Models are designed at different level of abstraction, for different purposes and in different **modeling formalisms**.

Models to communicate
amongst stakeholders

Models to calculate
performance indicators

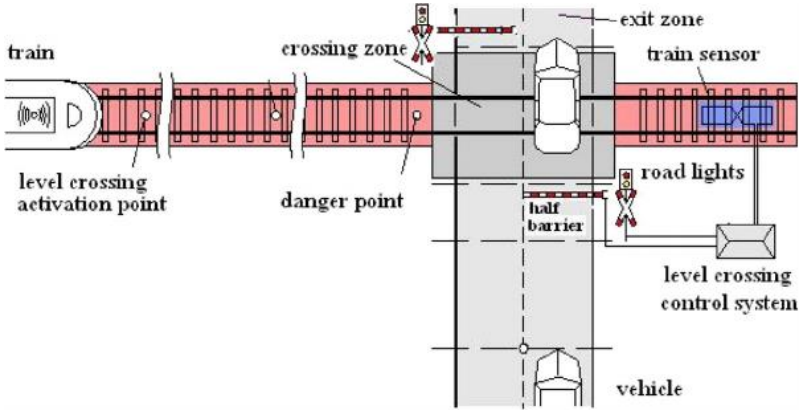
Models to generate artefacts (via code generation) or physical components (via additive manufacturing)

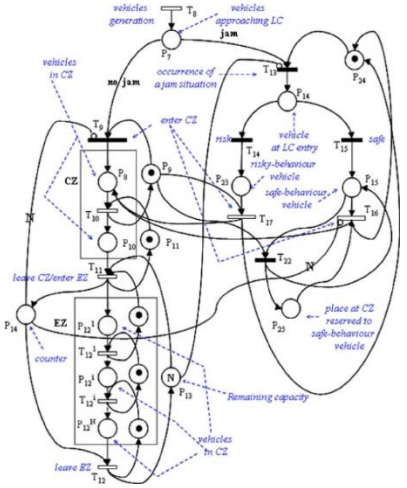
Informal models, even thought they are written in *standardized notations*, sometimes called *semi-formal* and computerized

Formal models, that essentially encode and organize (a given type of) mathematical equations

Discrete Mathematics

The incredible successes of sciences, the fantastic development of technologies rely for a great part on mathematical analysis, i.e. mathematics of the continuum. However, when considering a system (a car, an airplane, an engine…) **at system level**, it turns out that **relevant conceptual tools** come rather from **discrete mathematics**.





To reason about a system, it is convenient to see it as a **finite hierarchy of components** and to consider that each component can be in **finitely many states**. As a consequence, the number of possible states of the system itself is also finite, although it can be extremely large. The **modeling frameworks** we shall study in this course are **discrete**.

Complexity of calculations

The **value of a model** stands entirely in the lessons one can draw on the system from the **virtual experiments** made on the model. These virtual experiments have a **computational cost**. There are cases in which this cost is actually the decisive factor.

The rules of chess are fully known and fully deterministic. Moreover, a game can end only in three ways: white wins, black wins or draw.
It is easy to design a model of what is the initial position, what is a legal move, and what is a final position.
However, determining whether there is a winning strategy for white, i.e. a way of playing white that wins against any defense of black, is computationally so expensive that no virtual experiment can answer the question.

CentraleSupélec

LECTURE 1. PART 1. SECTION 3. OBJECTIVES AND ORGANIZATION OF THE COURSE

Knowledge base of engineers

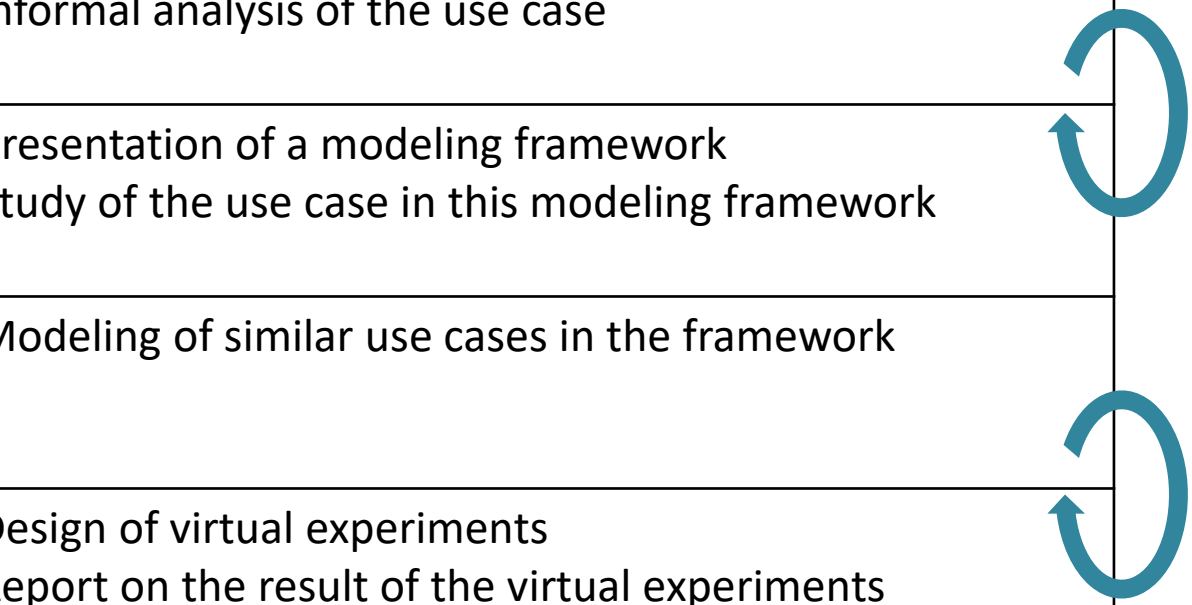
As future engineers and scientists, we should acquire the **scientific knowledge** and **engineering know-how** making you able:

- To select **modeling framework and tools** to solve problems at stake, meeting needs and constraints of your work environment;
- To **design** models;
- To understand the **representativeness** and the **limits** of these models;
- To perform **virtual experiments** with these models, defining appropriate **experimental protocols**;
- To **interpret results** of these experiments and to check them against the real world;
- To **tune** models accordingly;
- To **manage** models and virtual experiments**;**
- To **integrate** your models and virtual experiments with those of other teams/disciplines/organizations.

Organization of the course

We shall try to follow the following scheme for lectures and the tutorial.

lectures	Observe	• Description of a use case • Informal analysis of the use case
	Formalize	• Presentation of a modeling framework • Study of the use case in this modeling framework
tutorials	Model	• Modeling of similar use cases in the framework
	Experiment	• Design of virtual experiments • Report on the result of the virtual experiments



CentraleSupélec

Domain Specific Languages

Modeling activities in this course (and the associated tutorials) rely on:

1. Small **textual domain specific modeling languages**;
2. Ad-hoc **assessment tools** for models written in these modeling languages.

Assessment tools have been written in the **Python** scripting language.

You should download python and install it on your machine.

Please download and install the standard Python 3.5.x environment.

https://www.python.org/downloads

CentraleSupélec

Scripting Languages

Programs play a central role in all engineering processes (and more generally in modern societies). Therefore, systems engineering has much to do with computer science and software engineering. However, this relationship must be well understood.

- **Computer Science** is the science of **discrete structures** (as opposed to mathematical analysis). It provides many concepts (automata, languages, computational complexity…) used in other sciences (including cognitive sciences) and of course in Systems Engineering.
- **Software Engineering** is a high technology that aims at designing methods to develop software. It is only marginally related to Systems Engineering.
- **Scripting** (with programming languages such as Perl, **Python**, Ruby) is a particular type of programming. Scripting is mandatory to be efficient when performing virtual experiments (e.g. to change formats of data, to chain tools…).

Mastering at least one scripting language is one of expected skills of new generations of engineers and scientists.

CentraleSupélec

Content of the course

11 lectures + 11 tutorials + one written exam

Introduction

- Lecture 1, part 1: *The science of models*

Models to communicate

- Lecture 1, part 2 + tutorial 1: *Business Process Model and Notation*
- Lecture 2 + tutorial 2: *System Architecture & Design Structure Matrices*
- Lecture 3 + tutorial 3: *System Architecture & Requirement Engineering*

Models to calculate

- Lecture 4 + tutorial 4: *Graphs*
- Lecture 5 + tutorial 5: *Automata*
- Lecture 6 + tutorial 6: *Constraint Solving*
- Lecture 7 + tutorial 7: *Optimization* 
- Lecture 8 + tutorial 8: *Discrete Event Simulation*
- Lecture 9 + tutorial 9: *Stochastic Simulation*
- Lecture 10 + tutorial 10: *Machine Learning* 
- Lecture 11 + tutorial 11: *Limits of Calculability*

Assignements

1. Send pdf versions of your reports (to avoid compatibility issues).
2. Name reports according to the following rule.
   - «TPK5120 – 2016-2017 - Doo John - Assignment 1 - version 2.pdf »
3. Start your report with a header with your first name, your family name, the number of the assignment, the date…
4. Organize your document with sections, subsections, table of content and the like. Use styles (Word/LaTeX/…) to do so.
5. Write in good style, preferably in English. Norwegian and French are accepted as well.
6. Be efficient. All relevant information should be in your report, but no more.
7. Describe carefully experimental protocols (tested hypotheses, experiments, results, interpretation of the results).
8. …

How is the course evaluated?

Two grades will contribute to your final grade:

- A writing exam (50%).
- A grade obtained from the 3 assignments you are asked to submit (50%).

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.