



Elements of System Dynamics

Antoine B. Rauzy

PART 1. INTRODUCTION

LECTURE 1. SYNOPSIS

LECTURE 1. SYNOPSIS

Key notions:

- Systems
- System Dynamics
- Models and modeling frameworks
- Experiments in silico
- Systemic Digital Twins

Agenda

Section 1. System Dynamics

Section 2. Systemic Digital Twins

Section 3. Σ^{TM} and WorldLabTM

Section 4. The Σ^{TM} Modeling Framework

Section 5. Learning Objectives

Section 6. Agenda of the Course

Section 7. Collective Exercise

LECTURE 1. SYNOPSIS

SECTION 1. SYSTEM DYNAMICS

What Is a “System”?

- 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 systems**, i.e. systems designed by means of an engineering process. We are thus interested primarily in:
 - Industrial **products** ;
 - **Processes** by which organizations design, produce, operate and/or regulate these products.
- We shall focus less on:
 - Natural systems (molecules, cells, volcanos, rivers...)
 - Human systems not resulting of an engineering process (cities, internet...)
 - Software systems.

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

System Dynamics

System dynamics is a methodology for understanding, modeling, and simulating complex systems, typically aimed at addressing complex problems with **interdependent factors**, **feedback loops**, and **time delays**. Developed by **Jay W. Forrester** in the late 1950s at the **MIT Sloan School of Management**, system dynamics uses **differential equations** to capture and simulate the **behavior of systems over time**, allowing for the examination of interactions between components within economic, environmental, and organizational systems.

System dynamics is widely known for its application in fields such as **business management**, **environmental studies**, and **public policy**. Models are often built with **stock and flow diagrams** and causal loop diagrams, which help in identifying feedback structures within systems.

Why a Course on System Dynamics?

“Don’t learn economics, learn system dynamics”.

Steve Keen, Head of the School of Economics at Kingston University in London

By learning **system dynamics**, you learn tools to understand and model **complex systems**, an essential skill in today’s interconnected world. This approach emphasizes **systemic thinking**, allowing to explore how parts of a system interact over time and how feedback loops and delays impact overall behavior. Such skills are applicable **across diverse fields**—economics, environmental science, engineering, healthcare, and public policy.

By learning system dynamics, you can better anticipate unintended consequences of decisions, model complex problems, and test potential solutions in a simulated environment before implementing them in real life. Moreover, foundational concepts like **feedback**, **stocks and flows**, and **non-linear dynamics** deepen their analytical skills, making you better at **problem-solving**.

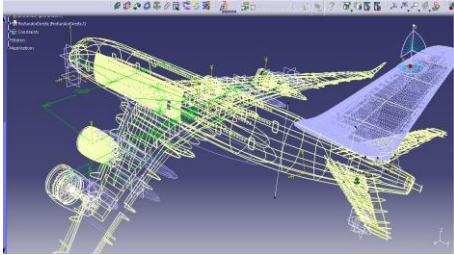
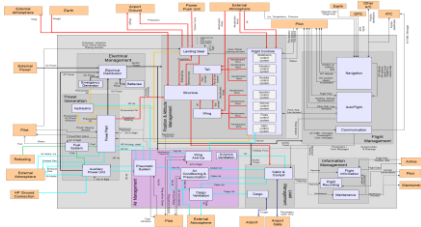
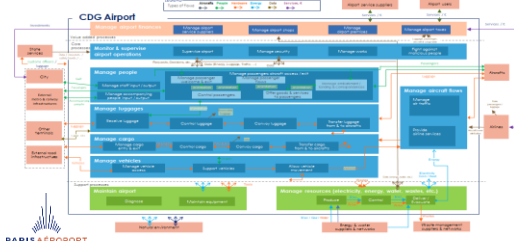
LECTURE 1. SYNOPSIS

SECTION 2. SYSTEMIC DIGITAL TWINS

Data and Models Are Pervasive in Engineering

- 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.
- To face this complexity, the different engineering disciplines (mechanics, thermic, electric and electronic, software, architecture...) largely virtualize their contents, i.e. they are managing more and more **data**, and they are designing more and more **models**. Data without models to interpret them are useless. We entered the era of **model-based systems engineering**.
- We call the sets of artifacts (data + models) that are used to design, produce, operate, and decommission systems the **digital twins** of these systems.

Systemic Digital Twins

	Industrial products	Industrial processes
Geometry	 A 3D wireframe model of an aircraft, showing the internal structure and components, rendered in a yellow and green color scheme. <i>Example: geometric model of an aircraft</i>	 A 3D perspective view of an airport terminal and runways, showing the layout of the facility and the presence of several aircraft. <i>Example: geometric model of an airport</i>
Behaviour	 A complex functional block diagram of an aircraft, showing the interconnections between various systems and components, including engines, wings, and landing gear. <i>Example: functional model of an aircraft</i>	 A detailed flowchart illustrating the business processes of an airport, including check-in, boarding, baggage handling, and ground services, with the Paris Aéroport logo at the bottom. <i>Example: business processes of an airport</i>

Systemic digital twins gather **data and models** that used to **simulate** and to **optimize** complex industrial products and processes.

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 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”

CHAPTER 1. SYNOPSIS

SECTION 3. THE Σ^{TM} LANGUAGE

Σ^{TM} Language

This course relies on the Σ^{TM} **modeling language** for designing systemic digital twins of the systems under study.

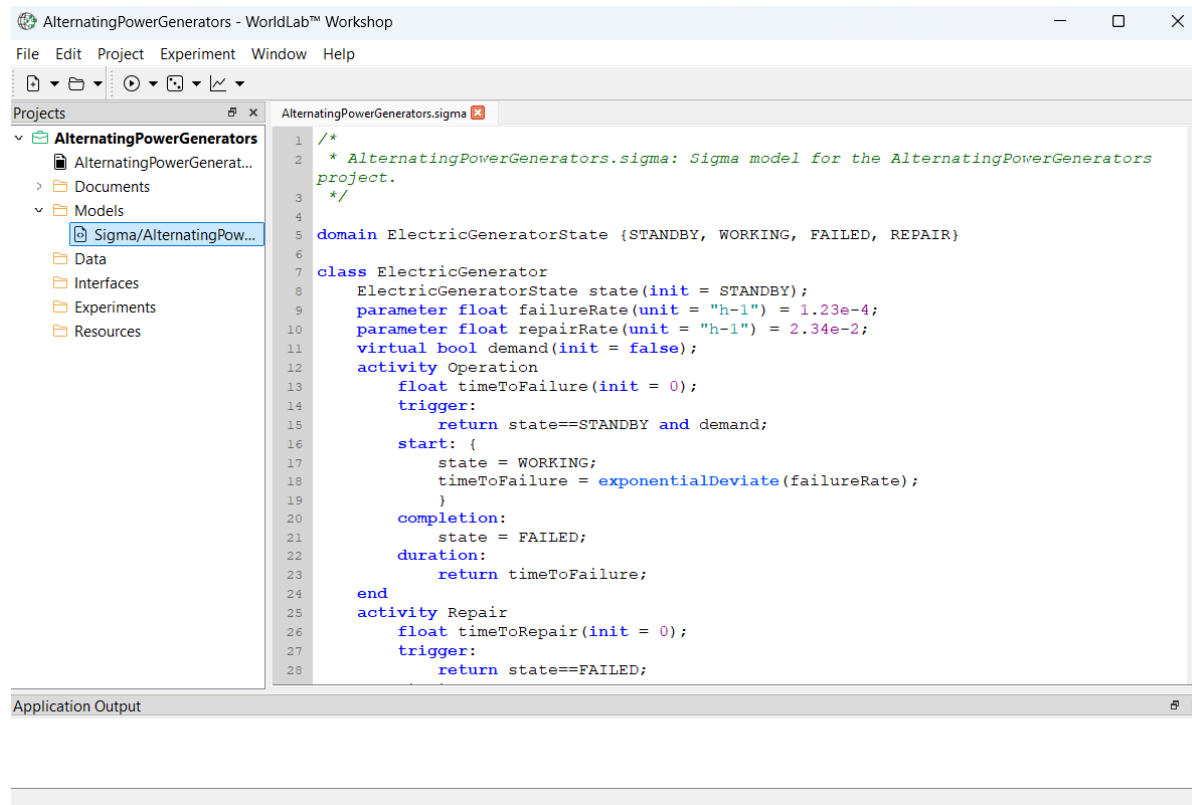
Σ^{TM} is an **object-oriented** textual modeling language. It makes it possible to describe both the **hierarchical structure** of the system under study and its **behavior**. Σ^{TM} is **agnostic** in this sense that it is not dedicated to any specific application domain.

Conversely to modeling frameworks “traditionally” used in system dynamics, which rely on differential equations, Σ^{TM} relies on **discrete event simulation**. This mathematical framework is much more appropriate than differential equations to represent the behaviors of complex systems.

Σ^{TM} makes it possible to represent both **deterministic** and **stochastic** behaviors as well as to embed **external data**.

WordLab™ Workshop

The **WorldLab™ Workshop** is an **integrated modeling and simulation environment** supporting the development of systemic digital twins using the Σ^{TM} modeling language.



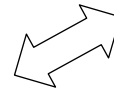
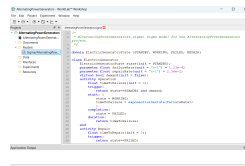
WordLab™ Workshop

Languages:

- Σ^{TM} : core modeling language
- WIDL: graphical simulation interfaces
- Tau: documentation (LaTeX-like)

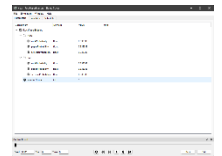
WorldLab™ Wizard

- Project management
- Model Authoring
- Tool launching



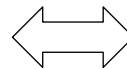
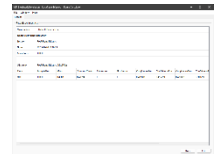
Interactive Simulator

- Model validation
- What-if scenarios



Stochastic Simulator

- Assessment of key performance indicators



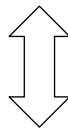
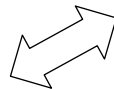
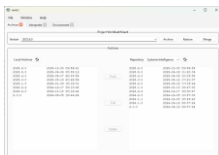
Analyst

- Post-processing of simulation results



wvmc

- Version management



Tau compiler

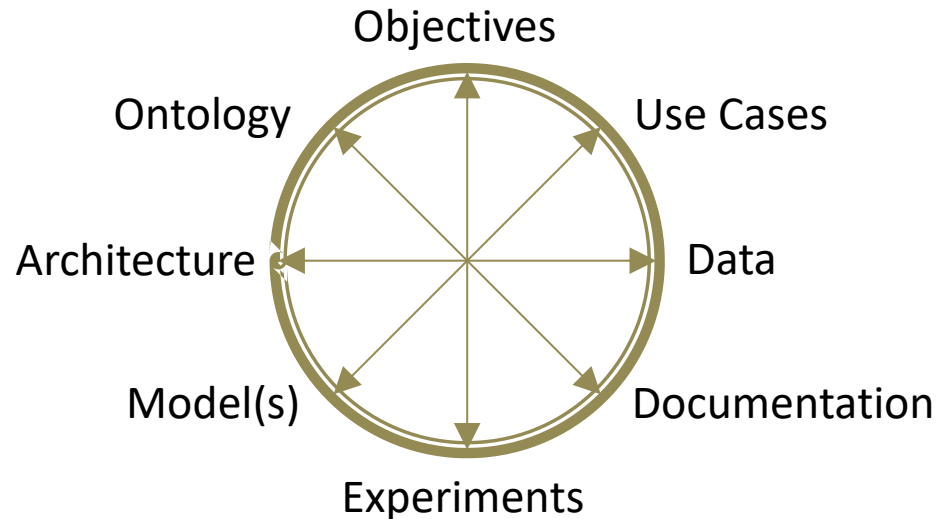
- Generation of PDF and HTML documents

CHAPTER 1. SYNOPSIS

SECTION 4. Σ^{TM} MODELING FRAMEWORK

Σ^{TM} Modeling Framework

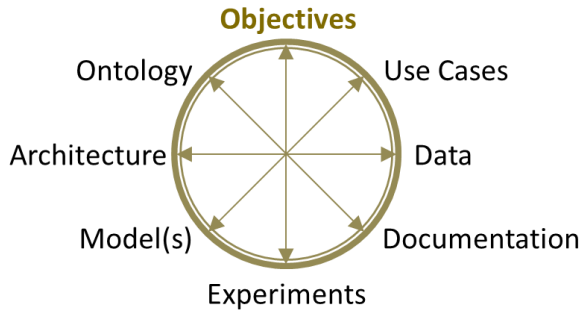
The Σ^{TM} **modeling framework** helps to design systemic digital twins efficiently and accurately. It consists of **activities** to be performed resulting in as many **artifacts**:



The analyst **must adapt** the Σ^{TM} modeling framework to each specific study. It is a general **guideline**, not a strict procedure to follow.

The process is **iterative**, the Σ^{TM} modeling framework is a **logical organization**, not a chronological process.

Objectives



Developing a systemic digital twin is a resource consuming, committing process. If one develops a systemic digital twin one must have good reasons to do so. It is thus of primary importance to **state**, with as few ambiguities as possible, what are the **objectives of the study**.

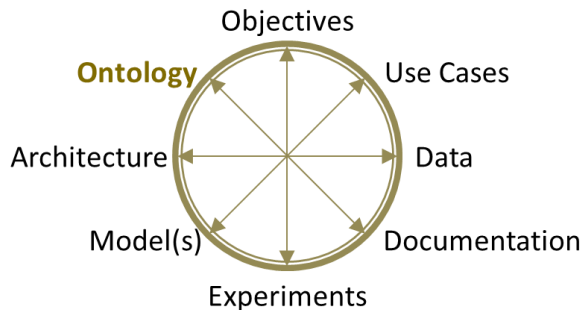
A systemic digital twin is developed by an **analyst** for a **client** (who is not a specialist of system dynamics). The objectives of the study, as stated by the analyst, must be **understood** and **agreed** by the client.

Key questions:

- What is the **system** of interest?
- What are the **issues** to be looked at?
- *What do you want to **learn** about the system?*

It is often a good to design small, **metaphorical models**, to illustrate the objectives.

Ontology

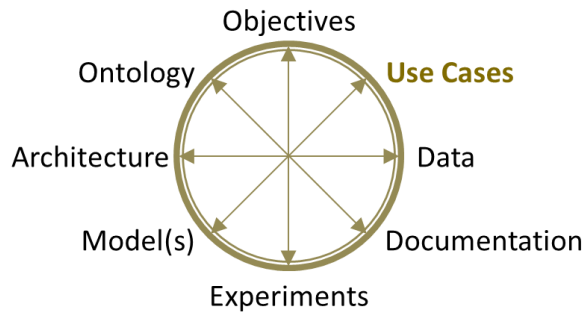


“In information science, an **ontology** encompasses a representation, formal naming, and definitions of the categories, properties, and relations between the concepts, data, or entities that pertain to one, many, or all domains of discourse” [Wikipedia].

To a design of a systemic digital twin, an ontology must at least contain:

- A **glossary** defining of all **technical terms** used in the domain of the system of interest.
- A **conceptual data model** including:
 - A description of core **entities** of the system and its environment;
 - A description of **relationships** existing between these entities;
 - A description of **business rules and constraints** that apply on entities and their relationships

Use Cases



"In both software and systems engineering, a **use case** is a structured description of a system's behavior as it responds to requests from external actors, aiming to achieve a specific goal" [Wikipedia]

External systems, whether they are human beings or not, are often called **actors**.

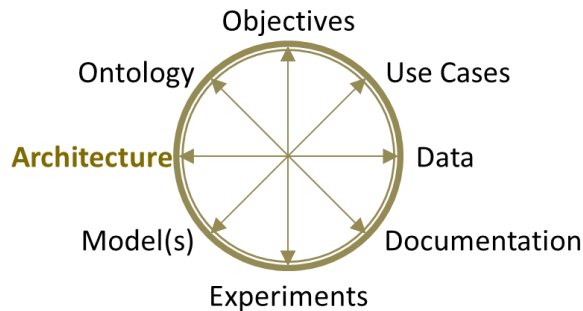
Use cases have three main roles:

1. They avoid **abstract non-sense** by making things very concrete.
2. They cover the main **scenarios of use** of the system.
3. They prepare **tests** to be performed on the system, once built.

In the case of systemic digital twins, the analyst may design actually two series of use cases:

- Use cases describing the **behavior** of the **system** of interest, and
- Use cases describing the **scenarios** of use of the **digital twin** itself.

Architecture



Designing the ontology of the system of interest provides a preliminary view on this system. The question now is how this system is structured and how it evolves through the time? In other words, what is its **structure** of the system and what is its **behavior**?

A technical or socio-technical system can be seen as a hierarchical network of human and technical **actors**. This network is called the **logical architecture** of the system.

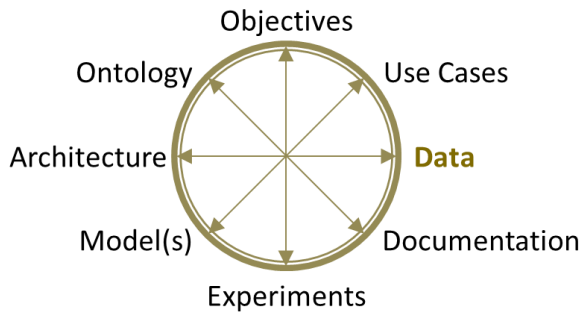
The global behavior of the system results from the **actions** of these actors. These actions results in general from **activities** themselves organized into **processes**. The system has thus also a **process architecture**.

The **stocks and flows** of matter, energy and information link the logical and the process architectures.

Studying the **dynamics** of a system means describing the actors, activities and processes at work in this system as well as its stock and flows.

“System” must be understood here in a broad sense: some actors and activities to be considered may belong to the **environment** of the system. Similarly, the concept of activity includes not only planned activities, but also hazards, changes in the environment...

Data



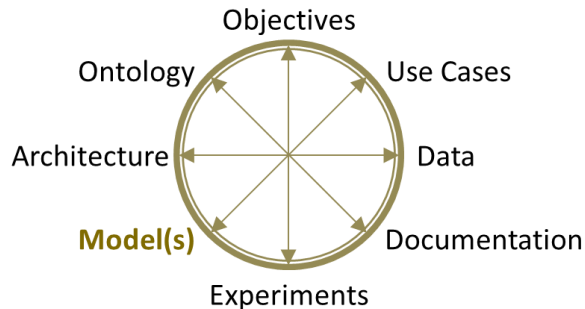
Data are a collection of discrete or continuous values that convey information, describing the quantity, quality, fact, statistics, other basic units of meaning, or simply sequences of symbols that may be further **interpreted formally** [Wikipedia].

The design and operation of any complex technical or socio-technical system requires and produces a lot of data. So does the design and operation of a systemic digital twin of the system.

Most of the times, these data are not gathered in a single place and ready to use. A significant part of the effort required to develop a systemic digital twin consists in **collecting, analyzing** and **formatting data**.

Complex systems are often subject to **uncertainties**. In systemic digital twins, these uncertainties are typically captured using **time series** and **probability distributions**. These are important data to collect, analyze and format.

Models



Σ^{TM} models aim at capturing the structure and the dynamics of the systems under study.

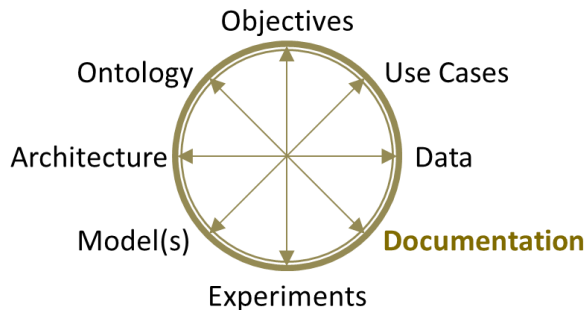
However, Σ^{TM} models do not aim at mimicking the physical behavior of the systems.

Models are **abstractions**. An abstraction is relevant if and only if it captures a relevant feature. The objective of designing a model is not to match the reality, but to **learn** something about the system under study.

The design of a model is **highly iterative**. Each iteration consists in adding a feature to the model, then to test it extensively. The smaller the iterations, the most efficient the design.

The development of a systemic digital twin requires in general the design of several Σ^{TM} models. It is in particular a good idea to **test features separately**, within a dedicated small models.

Documentation



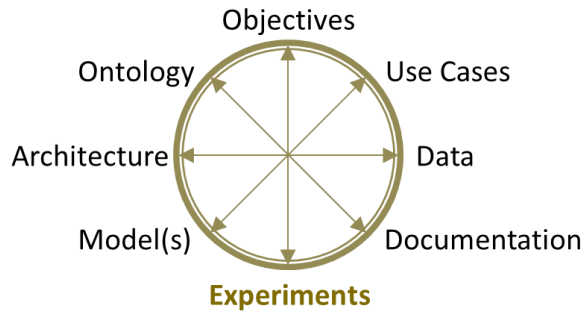
A systemic digital twin must be self-contained. It should integrate all the **artifacts** (documents, data, programs, models...) required for its design, as well as all artifacts it produces throughout its lifecycle.

Collecting and organizing the documentation of a system digital twin is essential. Recall that the main objective of a systemic digital twin is to learn about the system under study and to **transmit** and **capitalize** the acquired **knowledge**. Without a solid documentation this is just impossible.

The adage “*a picture is worth a thousand words*” rightly highlights the expressive power of images, diagrams, and other graphical representations. Nevertheless, when the objective is to formulate a precise, unambiguous, and formally verifiable description, written language (text) remains the indispensable medium.

Last but not least: a good documentation is a **concise** documentation. The elegance is reached when everything unessential has been removed.

Experiments



In system dynamics, **experiments** consists mainly in performing **interactive** or **stochastic simulations**. These are experiments **in silico**.

Experiments in silico are nevertheless experiments. As such they must obey **rigorous experimental protocols**:

- Hypotheses must be clearly set up.
- Operations to perform must be clearly defined.
- Conclusions drawn for the system must follow from the results of experiments on the model.

Moreover, they must be (easily) **reproducible**.

Concretely, simulations of Σ^{TM} models involve two types of **parameters**: parameters of the model itself and parameters of the simulation. Both must be carefully managed for the experiment to be of interest.

CHAPTER 1. SYNOPSIS

SECTION 5. LEARNING OBJECTIVES

Learning Objectives

The objective of the course is to make you able to:

- Understand the **key features of a system**, what are the **problems at stake** and which **key performance indicators** must be calculated.
- Look for **documentation** on the system (scientific articles, technical documents, relevant data) and synthesize it.
- Design a Σ^{TM} **model** for the system using the Σ^{TM} **modeling framework**.
- Beyond the Σ^{TM} model itself, to develop a whole **systemic digital twin** using the **WorldLabTM Workshop** integrated modeling and simulation environment.
- Understand the **representativeness** and the **limits** of your models.
- Perform **experiments in silico** (simulations) on your models, defining appropriate **experimental protocols**.
- **Interpret results** of these experiments.
- **Manage** models and experiments in versions and configurations.

In a word, this course aims at giving you a both a fundamental and practical knowledge on **system dynamics** and **systemic digital twins**.

A Multidisciplinary Approach

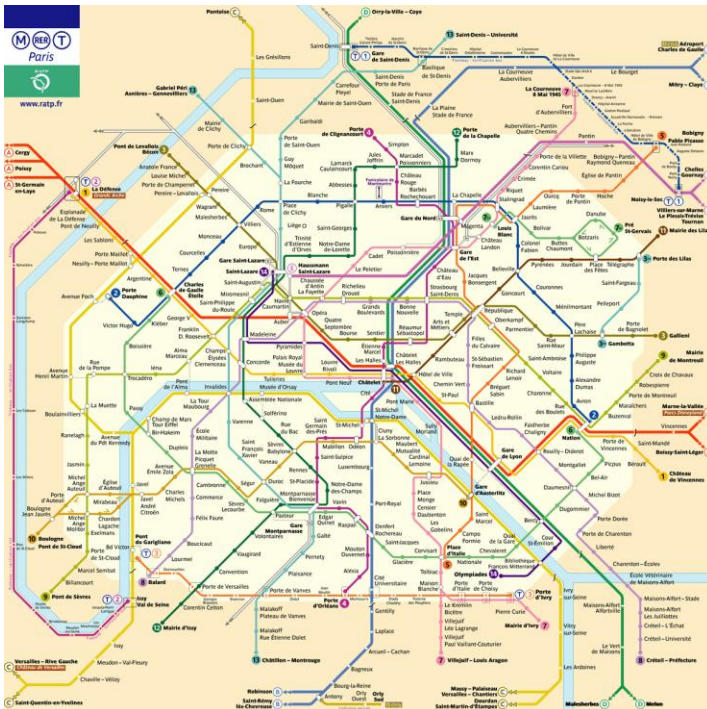
This course is tightly linked with my scientific research activities. I foster here is a multidisciplinary approach. The course involves concepts, methods and tools from system dynamics, software engineering, system engineering, reliability engineering, but also from computer science, linguistic, cognitive sciences, management science and a few other. E.g.

Metaphorical models: in linguistic, a **metaphor** is a figure of speech in which one concept is described in terms of another, highlighting similarities between them, e.g. “*Time is a thief*”. Metaphors are powerful tools for simplifying complex ideas, creating vivid imagery, and enabling deeper understanding. **Metaphorical models** are small models that are used to demonstrate a key idea.

Simplexity: Introduced, among other by the neurophysiologist A. Berthoz, the concept of **simplexity** describes the mechanisms by which the (human) brain manages the complexity of the environment. These mechanisms do not eliminate the complexity but rather tame it via heuristics. Behavioral modeling of complex systems is a typical application of this concept in engineering.

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 specific **purpose**, i.e. to realize specific **computer-based experiments**, to **learn lessons** from these experiments and to make (hopefully better) **decisions** about the system.

Learning Material

Course Material:

- Slides of the course
- WorldLab™ Workshop documentation
- Σ^m Book (*parts that are ready*)

Relevant literature (recommended):

- Donella H. Meadows, Jorgen Randers and Dennis L. Meadows. Limits to Growth: The 30-Year Update. Chelsea Green Publishing. ISBN 978-1931498586. 2004.
- Donella H. Meadows. Thinking in Systems: A Primer. Chelsea Green Publishing. ISBN 978-1603580557. 2008.
- John D. Sterman. Business Dynamics: Systems thinking and modeling for a complex world. McGraw Hill. New York, NY, USA. ISBN 978-0-07-231135-8. 2000.

CHAPTER 1. SYNOPSIS

SECTION 6. AGENDA OF THE COURSE

Agenda of the Course

Part I. Introduction

- Lecture 1. Synopsis
- Lecture 2. Getting Started

Part II. The Σ^{TM} language

- Lecture 3. Diving Deeper into Σ^{TM}
- Lecture 4. Architecture of Models
- Lecture 9. Advanced Topics

Part III. The WorldLabTM Workshop

- Lecture 5. Experiments
- Lecture 8. Interfacing and Documenting Models

Part IV. Systemic Digital Twins

- Lecture 6. Model-Based Systems Engineering
- Lecture 7. The Σ^{TM} Modeling Framework
- Lecture 10. Reliability Engineering
- Lecture 11. Scripting and Optimization
- Lecture 12. Computational Complexity Issues

How is the Course Evaluated?

The grade is obtained from the 4 (mini) projects.

- Projects can be done in small groups (maximum 3 persons).
- All projects are mandatory.
- Each project is graded on 25 points (for a total of 100 points).

Projects for 2026 (tentative)

Deadlines

- | | |
|---|------------|
| • Project 1. Insurance Company | 13/02/2026 |
| • Project 2. Ship Propulsion System | 20/03/2026 |
| • Project 3. Wafer Production Line | 17/04/2026 |
| • Project 4. Plant Construction Project | 15/05/2026 |

Deadlines are strict. You can submit preliminary versions to get feedbacks.

Projects: How (take 1)?

1. Each project consists in designing a Σ^{TM} model and to perform experiments (simulations) on this model.
2. For each project, you must submit your model, a thorough presentation of the how you designed it, of the experiments you have performed on this model and a discussion on the results of these experiments.
3. For the first project(s), you shall submit a separated report, in pdf. For the last ones, this report must be integrated into your digital twin and written in Tau.
4. Write in good style, preferably in English. Norwegian is accepted as well. Be efficient. All relevant information should be in your report, but no more.
5. Organize your document with sections, subsections, table of content and the like.

Projects: How (take 2)?

6. Projects must be submitted in Blackboard as zip or tar.gz archives and named according to the following rule.
«TPK5120 – 2025 – Group Name - Assignment 1 - version 2.pdf »
(Choose the name of the group you like).
7. Reports must start with a header indicating with your first names, your family names, the number of the assignment, the date...

Warning: I will not correct projects not submitted according to the above rules.

Important Advices

- This course is quite demanding. Work continuously on it.
- This course requires some mathematical and programming skills. Don't be afraid. Reason "from first principles".
- Modeling is like programming which like swimming, which is like playing the piano, which is... you can't learn it just by reading the course (or even books). You have to practice, practice and practice again.
- Modeling is a difficult task. Don't expect to get the result straight away. Rather, proceed by small incremental steps: write a small, incomplete model first, then add progressively features, testing your model at each step.
- The use of AI to learn about systems under study is ~~authorized~~ recommended... with the usual warnings.

CHAPTER 1. SYNOPSIS

SECTION 7. COLLECTIVE EXERCISE

E-commerce Marketplace

ShopHub is a fast-growing European online marketplace:

- Customers buy goods sold by independent vendors
- Payments are made to ShopHub, vendors are paid later
- Shipping may be done by the vendor or ShopHub fulfillment centers
- Real-time stock accuracy is essential

A viral influencer announces a flash sale on select items. Traffic increases by a factor 10.

Orders spike, causing:

- Stockouts
- Conflicting orders for the same inventory
- Shipment delays
- Increased cancellations and customer support load

ShopHub wants to improve resilience of its order management and fulfillment processes. You are asked to help them with that.