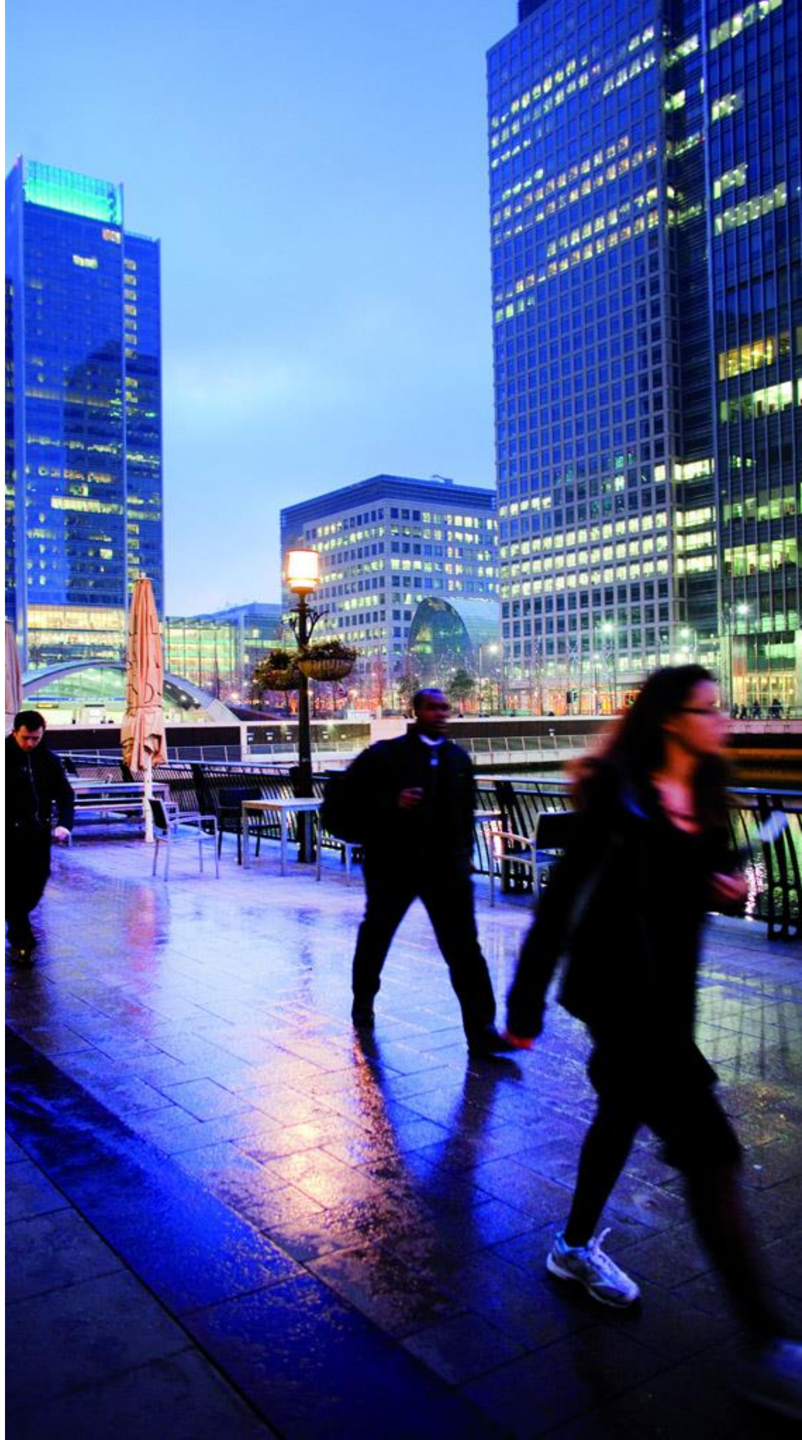
Requirements Engineering for Cyber-Physical Systems

And also for Socio-Technical Systems and Systems of Systems

Thuy NGUYEN
thuy.apt[at]orange.fr

GDR GPL, IE
October 5th, 2023
Sophia Antipolis, France



Who Am I?

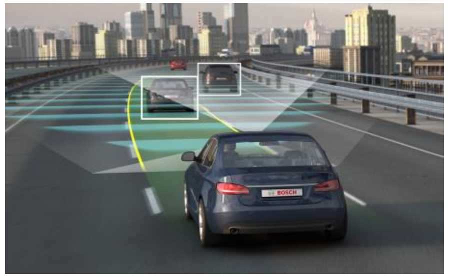
- **1975 - 1994: Software engineer and architect in the general software industry**
  - Signal acquisition & processing
  - Programming languages, compilers & interpreters
  - Computer graphics, computer-aided industrial drawing, mechanical CAD-CAM
  - Real-time, distributed digital systems
  - File & database management systems
  - Software engineering

- **1994 - 2021: Research engineer at EDF for Instrumentation & Control (I&C) systems important to power plant safety**
  - Since 1994: formal verification (complete I&C system software, and I&C system architectures)
  - Since 1999: FPGAs (Field Programmable Gate Arrays) for safety applications
  - Since 2007: simulation assisted engineering of cyber-physical systems, socio-technical systems and systems of systems
  - Since 2016: NUWARD I&C architect
    - NUWARD is the SMR (Small Modular Reactor) co-developed by EDF, CEA, Technicatome and Naval Group

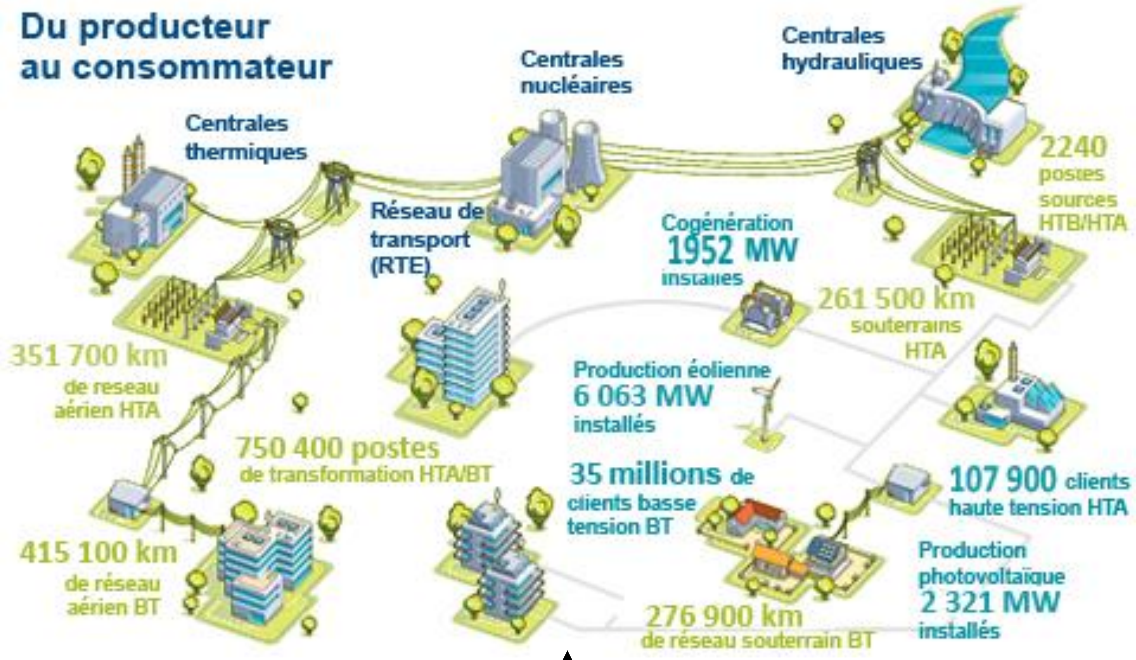
- **Since June 2021: Retired**
  - But still active

Cyber - Physical Systems (CPS) , Socio - Technical Systems (STS)

- **Computation & networking**
- **Physical processes, physical proximity, physical connections, ...**
- **Human and organisational aspects**



Cyber aspects need to be addressed in the framework of human, physical and overall system aspects



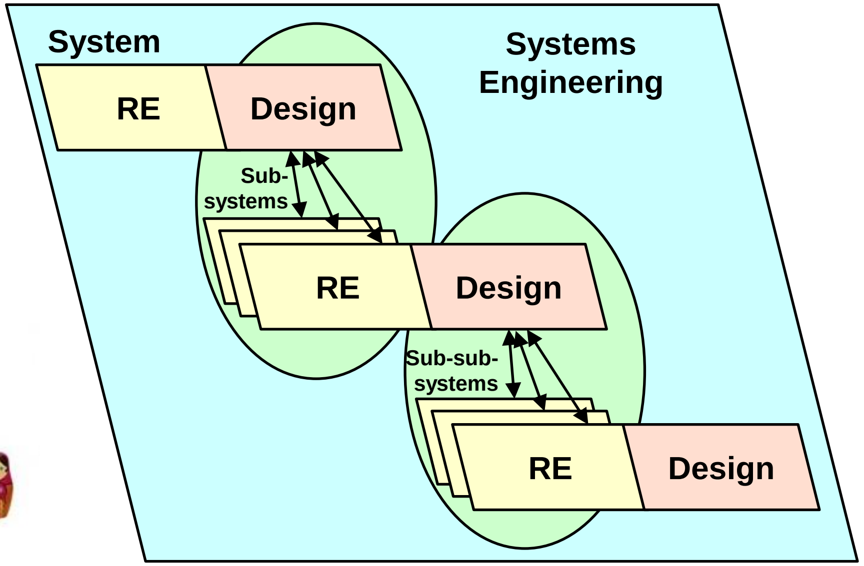
Systems of Systems (SoS)

bRSE

- **Often, requirements engineering (RE) is considered as a mere phase in the systems engineering (SE) process**
  - Different and separate from the design and implementation of solutions

- **For large and complex CPS, this cannot be so**
  - They are recursive
    - Subsystems are often full-fledged systems of their own
    - The design of a system consists of RE for its subsystems
  - RE is necessary for difficult and complex activities all along system life cycle
    - Construction, installation on site, operation, maintenance, modernisation, deconstruction
    - Airbus' MOFLT (Missions to Operational, Functional, Logical and Technical elements) approach
- **RE for CPS involves many participants having their own viewpoints and expectations on the system, and their own engineering methods and languages**
  - Teams in charge of subsystems, engineering disciplines, organizations, stakeholders
- **RE needs to be informed by other engineering activities**
  - Such as cost and feasibility studies ➔ Methods and languages need to be integrated

➔RE is inextricably intermingled with SE: they cannot be separated
  - RSE (Requirements and Systems Engineering)
  - bRSE is the part of RSE that applies to CPS-STS dynamic and behavioural aspects



Developers (Maîtres d'Oeuvre - MOE)

- **MOE are responsible for the design and implemention of a system or sub-system**
  - Not of its operation
- **They receive user requirements (cahier des charges) as an input and consider them as their starting point**
  - They look for possible defects, essentially as impediments to their own work
    - "What does that mean?"
    - "Can I implement that?"

RSE is sometimes (often) '*hijacked*' by MOE

Owners (Maîtres d'Ouvrage - MOA)

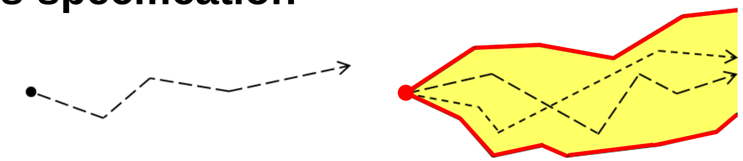
- **MOA are responsible for the system over its complete life cycle**
  - From initial conceptual studies to deconstruction
- **They have to elicit and specify high-level user requirements, and validate detailed technical requirements considering**
  - Possible consequences at each stage of the system life cycle
  - The various and numerous situations (normal and abnormal, internal or external) the system may face at each stage
  - The often contradictory viewpoints of numerous stakeholders
- **Defects could lead to unacceptable consequences**
  - Delays; Excessive cost in development, operation, maintenance; Catastrophic damage to property and/or the environment; Human death; ...
    - "Could that bankrupt my organisation?"
    - "Will that kill people?"
    - "Could that send me to jail?"
- **For MOA, the specification of requirements is a strategic, long and difficult process**

Defects in Specified Requirements (an MOA's Viewpoint)

- **Inadequacy**
  - Where, in some situations, what is specified is woefully inappropriate and could lead to unacceptable consequences
  - Or where what is necessary in some situations is not specified (silence), which could also lead to unacceptable consequences
- **Ambiguity**
  - Where different people concerned could understand what is specified differently, which could also lead to unacceptable consequences
  - Syntactic ambiguity, lexical ambiguity, value ambiguity, ...
- **Apathy**
  - Where what is specified makes no difference between what is genuinely needed and what is barely tolerated in exceptional situations
- **Over-ambition**
  - Where what is specified might be interesting but is not essential and could lead to excessive complexity, higher costs, longer delays and greater risks of errors (in design, construction, operation and / or maintenance), with possibly unacceptable consequences
- **Over-specification**
  - Where what is specified is not the problem but a technical solution, not necessarily the best and simplest, and worse, not necessarily fully solving the real problem
- **Intangibility**
  - Where what is specified is based on immaterial, abstract concepts, with no concrete, verifiable acceptance criteria (wishful thinking)
- **Infeasibility**
  - Where what is specified is not feasible
  - E.g., when satisfaction of some requirements necessarily implies violation of others (contradiction)

bRSE is Much More than Requirements Management (RM)

- **To eliminate such imperfections, one needs to consider the individual and collective meaning of the specified requirements**
  - Addressing not just form and appearance, but also intentions and semantics
- **To avoid over-specification, requirements specification should not be deterministic and executable ➔ Constraint-based formal requirements specification**



- **The adequacy of specified requirements depends on assumptions made regarding environment and operation**
  - In rigorous bRSE, assumptions are as essential as requirements
  - They are the two faces of the same coin: the requirements of one are often assumptions of others
  - Formal specification of assumptions enables automatic test case generation and is necessary for formal verification

bRSE as a Process

- **It always starts with imperfect requirements suffering from some or all of the aforementioned defects**
  - Not only that is generally inevitable, but often, that is desirable. Sometimes, that is even necessary
- **The objective of the bRSE process is to gradually correct these**
  - And also to keep track of improvements, when that provides useful insights

Natural language requirements: often ambiguous and incomplete but more apt at conveying true intentions	**formalisation** →	Formally specified requirements: less ambiguous and more complete but also less easy to understand
Requirements expressed in terms of high level but abstract and intangible concepts that can be neither observed nor precisely measured, and with no clear acceptance criteria, but that also are more apt at conveying true intentions	**concretisation** →	Association with solutions in the form of concrete, technical, verifiable requirements, that would be considered as over-specification had they been specified right at the beginning
Requirements conveying true intentions but that are eventually revealed to be over-ambitious, unachievable or inadequate in some situations	**substitution** →	Traceable replacement with more pragmatic, achievable, adequate requirements, but that might be less easy to understand and might not convey true intentions

bRSE Process for CPS

Requirements Verification

- **At each step of the bRSE process, defects in requirements need to be detected and amended**
- **For large CPS, the total number of subsystems, participants, engineering activities and situations is staggering ➔ Purely manual approaches are useful but not effective enough**
  - Much like for software, no one should be content just with reviews and inspections
- **Physical testing is extremely expensive, possible only late in the bRSE process, and sometimes very dangerous or outright impossible**

➔**bRSE needs to be supported by modelling, simulation and when possible by formal verification**
- And also by many activity-specific tools

➔**Behavioural requirements need to be specified and modelled in formal languages**

Modelling Modularity

- **There cannot be a single model, or even one model per participant, but series of interrelated and coordinated models, reflecting**
  - The step-by-step progress and refinement along system life cycle
  - The viewpoints of different participants
  - The needs of different activities
  - Possible alternative solutions
- **Each participant needs to focus on what is relevant to their activity on hand**
  - Leaving aside details that are unnecessary for that activity
- **With the help of well-defined interfaces and interactions**
  - Contracts for desired, engineered interfaces
  - Encroachments for undesired side effects due to proximity, connectivity, sharing of resources, ...

bRSE Process for CPS

Clarity

- **Requirements need not only to be rigorously specified, they must also be clear to all concerned participants**
  - Even though they are inefficient for verifying the detailed behaviours implied by requirements, inspections and reviews by domain experts are necessary to verify overall soundness
- **That applies to requirements expressed in natural language, but even more urgently to formally specified requirements**
  - Domain experts are generally not specialists of academic formal languages

Top-Down & Bottom-Up Approaches

- **No real-life CPS-STS is engineered in a pure top-down approach**
  - At some point, one will rely on existing, off-the-shelf products and solutions
  - They could be internal to the organisation in charge of the CPS or provided by external suppliers and contractors
- **bRSE must be able to exploit existing solutions and models as they are**
  - I.e., without having to modify them
  - Even when their owners protect their know-how by providing them in non-readable formats

CPS Specific Features

- **Time**
  - Generally, a single continuous (Newtonian) time domain
  - Possibly, multiple continuous (Einsteinian) time domains
  - Possibly, multiple discrete time domains
  - Everything (or nearly everything) proceeds in parallel
    - Not essentially sequentially like in software
- **Timing**
  - Timing margins are always necessary: *When event E, action A shall be performed* will not do
  - Too late often means failure: *After event E, action A shall eventually be performed* will not do either
- **Physical quantities and continuous states**
  - E.g., temperature or pressure
  - One always needs to specify physical units: *When pressure > 10 do A otherwise do B* will not do
  - Like for timing, one always needs to specify margins: *When pressure > 10 bars do A otherwise do B* will not do either
- **Variety of human interactions**
  - For normal operation, but also for construction, and after that, for activities such as operation, in-the-field inspections, testing and maintenance, and ultimately for decommissioning
- **Randomness**
  - Due to noise, variability of physical manufacturing, hardware failures, external events, human behaviour and errors
- **Non-engineered interactions**
  - Interactions result not only from engineered interfaces, but also from unwanted effects
  - Due e.g. to proximity (e.g., heating or electromagnetic interference) or connections (e.g., electric or pressure shocks)
- **Passive components and structures**
  - E.g., wires, pipes and connectors, walls and openings
  - They must be subject to requirements as they may affect behaviour
- **Long (very long) life times**
  - Often, years and decades. Some SoS are "immortal"

CPS Dependability - 1/3

- **Reliability is the probability that the system will operate without failure for a given time period**

  Objective goal$_1$ "..." ;

  Requirement pfd$_1$ "The probability of failure on demand of goal$_1$ shall be lower than 10^{-4}";

  Requirement fro$_1$ "The failure rate in operation of goal$_1$ shall be lower than $1/(10^4$ h)";

  Requirement sar$_1$ "The spurious actuation rate of goal$_1$ shall be lower than $1/(10^2$ year)";

  - Probabilistic requirements cannot be verified with individual test cases
  - They need analytical approaches (in very simplified cases)
  - ... or statistical approaches based on very large numbers of test cases
    - E.g., Monte Carlo testing

- **Availability is the percentage of time the system is or must be operational**

  Requirement avail$_1$ "The planned unavailability of the system shall be lower than 8%";

  Requirement avail$_2$ "The unplanned unavailability of the system shall be lower than 5%";

- **Maintainability is the probability that each necessary maintenance action can be successfully performed**
  - Within a stated delay
  - Within a specified cost
  - ...

CPS Dependability - 2/3

- **Safety is the ability of the system not to harm people or the environment**
  - It can be specified in terms of actions to be performed or states to be maintained, but also of required absence of action
    - For safety-critical systems, such requirements are always probabilistic
  - It can also be specified in terms of safety class
    - Placing deterministic requirements on system architecture and engineering process
    - The Boeing 737 MAX accidents were due in part by an inadequate safety classification of the MCAS (Manoeuvering Characteristics Augmentation System)
- **Security is the ability of the system to resist to intentional aggressions**
  - It is more art than science, but some aspects can be specified in terms of negated capability requirements
  - Or in terms of time and effort necessary for an attack to be successful
- **Fault-tolerance is the ability of the system to tolerate a certain number of internal errors or component failures**
  - As they are a strong driver for architectural design, fault-tolerance requirements are generally expressed early in the life cycle, and need to be formally specified at times when architecture, internal components and failure modes are not known yet
  - Single Failure Criterion: ability of the system to tolerate one initial component failure, and all its consequences, including failure propagation

CPS Dependability - 3/3

- **Ergonomics is the adequacy of human-system interfaces**
  - In particular (but not only) to enhance human efficiency and prevent and/or avoid human error
  - It may be specified in terms of probability of human error
  - It may also be specified in terms of abstract requirements (e.g., time and human effort to accomplish a given task) that are then refined into concrete technical requirements

- **Robustness is the ability of the system to tolerate beyond-design, non-intentional aggressions**
  - Which could for example be due to human errors or exceptional ambient conditions
  - It may be specified in probabilistic terms

- **Resilience is the ability of the system, in unforeseen or exceptional situations, to enable uses that can avoid or limit unacceptable consequences**
  - Though it is also more art than science, some aspects can be specified in terms of capability requirements

`Objective goal`$_2$ `"In situation X, the operator should be able to ensure condition C" ;`

  - It may also be specified in probabilistic terms

`Requirement resilience`$_2$ `"The probability of failure of goal`$_2$ `shall be lower than 20%";`

Conclusion

- **Most RE methods and languages developed for software engineering are not well-adapted to the RE and bRSE of CPS-STS**

- Requirements models
- Functional models
- Probabilistic models
  - Safety, availability, dependability ...
- Economic models
  - Costs & revenues
- Operational procedures
- Task scheduling models
- Geometric and topological models
- Process & multi-physics models
  - Thermodynamics, Electromagnetism, Fluid Mechanics, Kinematics, ...
- Business process models
- ...

***Behavioural modelling* (*dynamic phenomena*)**

BASAALT: Behaviour Analysis & Simulation All Along systems Life Time

FORM-L: FOrmal Requirements Modelling Language
(*One language to rule them all* ...)

Thank you for your attention

Any questions?