2SC5810 – Design and verification of critical systems

Instructors: Idir Ait Sadoune
Department: DÉPARTEMENT INFORMATIQUE
Language of instruction: FRANCAIS
Campus: CAMPUS DE PARIS - SACLAY
Workload (HEE): 60
On-site hours (HPE): 34,50

Description

This course aims to address both the design and verification of complex and critical systems using techniques from Software Engineering. As the components of such systems are heterogeneous (i.e. both continuous for physical and discrete for software), the methodologies and tools presented in this course will be multiple and integrated into a development cycle framework. The idea is to start the analysis phase by using semi-formal tools (UML, SysML,...), often used in systems engineering to describe the structure of the system and its interactions, then to address in a scientific way the design and validation phases using formal techniques of software engineering (timed, stochastic and hybrid modeling, temporal logic, model-checking). The main objective of such an approach is to show, through the formal models obtained, that the system does what is expected of it while respecting the constraints imposed by the specifications and by the environment, or in the opposite case, to extract the states of the system that may call into question its correct functioning. In the latter case, the economic gain is very interesting and appreciable by the engineers who can correct the problems detected by the verification of the model before going to the stage of the implementation (programming).

Quarter number
ST5

Prerequisites (in terms of CS courses)

- Information systems and programming
- Algorithmics and complexity
- Model representations and analysis

Syllabus

- Chapter 1 - Presentation of temporal logics: LTL, CTL (3h lectures and 3h tutorials).
- Chapter 2 - Verification with Model Checking (1h30 lectures and 3h tutorials).
- Chapter 2 - Timed Automata: Modeling and Verification (3h lectures and 6h tutorials).
- Chapter 3 - Stochastic Models: Modeling and Verification (3h lectures and 3h tutorials).
- Practical sessions (2 x 3h)

Class components (lecture, labs, etc.)

- 10,5h lectures
- 16,5h tutorials
- 6h Practical sessions

Grading

Final exam (1H30)

Resources

The contributors (speakers):

- Marc Aiguier, (Department of Computer Science)
- Idir Ait Sadoune, (Department of Computer Science)
- Paolo Ballarini, (Department of Computer Science)
- Lina Ye (Department of Computer Science)

Learning outcomes covered on the course

At the end of this course, the student will be able to:

- Model a critical software system by using different formal approaches (temporal logic, automata, timed automata, stochastic models, hybrid automata).
- Model a critical software system by taking into account different types of constraints (functional, non-functional, temporal, ...)
- Analyze in a scientific way the model of a critical software system using the techniques from Software Engineering (formal verification technique: model checking).
- Extract the states of a critical software system that can call into question its correct functioning.

- Validate the model of a critical software system (the system does what is expected of it).

Description of the skills acquired at the end of the course

- C1 - Analyze, design, and build complex systems with scientific, technological, human, and economic components
- C2 - Develop in-depth skills in an engineering field and a family of professions
- C4 - Have a sense of value creation for his company and his customers
- C6 - Be operational, responsible, and innovative in the digital world
- C7 - Know how to convince