INSA
INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
LYON

Software engineering & UML modeling: Introduction

Frédérique Laforest

WHO AM I?

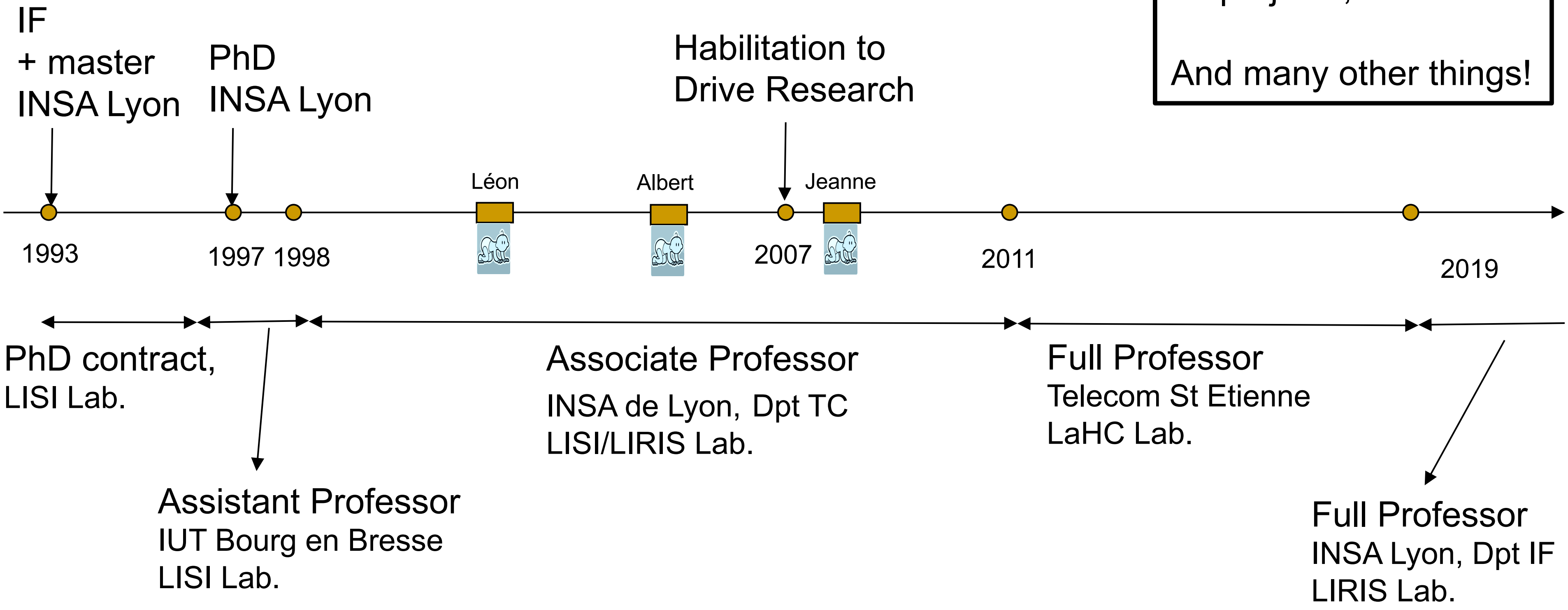
Frédérique Laforest

10 PhD students +3

Many papers

12 projects, 5 for EU

And many other things!



IF
+ master
INSA Lyon

PhD
INSA Lyon

Habilitation to
Drive Research

Léon

Albert

Jeanne

1993

1997 1998

2007

2011

2019

PhD contract,
LISI Lab.

Assistant Professor
IUT Bourg en Bresse
LISI Lab.

Associate Professor
INSA de Lyon, Dpt TC
LISI/LIRIS Lab.

Full Professor
Telecom St Etienne
LaHC Lab.

Full Professor
INSA Lyon, Dpt IF
LIRIS Lab.

Frédérique Laforest

- Full Professor in computer science
  - Research in LIRIS Lab., TWEAK team
    - Distributed reasoning,
    - Graphs streams querying & complex event processing,
    - Information extraction on the Web
  - Teaching at INSA Lyon
    - Software engineering & agile methods
      - IST, IF, Specialized Master in CS
  - Director of INSA Long Life Training

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

coswot

Focus on a research project

CoSWoT

Constrained Semantic Web of Things

ANR-19-CE23-0012

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

INRAE la science pour la vie, l'humain, la terre

LABORATOIRE HUBERT CURIEN UMR • CNRS • 5516 • SAINT-ETIENNE

LIRIS

MINES Saint-Étienne

MONDECA

Objectives of CoSWoT

CoSWoT will propose a **distributed WoT application platform** able to:

(1) use **graph-based knowledge models** to declaratively specify

- the semantics of the messages exchanged among the actual hardware nodes
- the domain knowledge and execution logic of a WoT application

(2) **distribute and process reasoning tasks** among heterogeneous nodes, including constrained devices by taking into account

- the hardware infrastructure
- the device characteristics

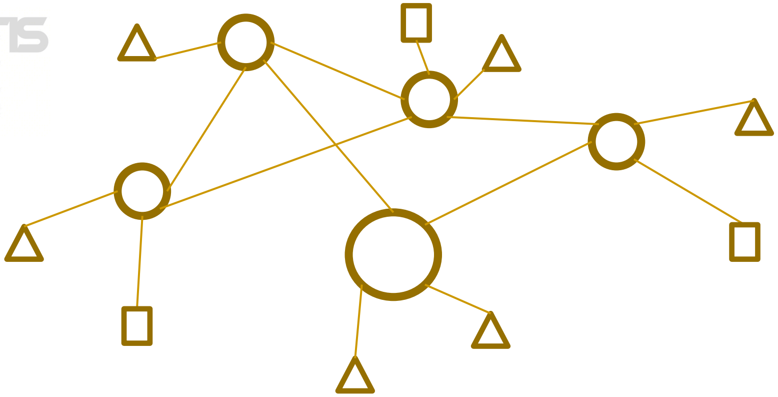
Our platform will enable the development and execution of decentralised smart WoT applications despite the heterogeneity of devices

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

Scenario – Smart Building

Campus of LyonTech-Doua and Plateforme Territoire Saint Etienne

- Integration of mobile sensors and actuators like smartphones and robots
- Personalized sensor data reporting
- User comfort/care/security requirements



Validation of foundations
- Interoperability
- service discovery
- data velocity

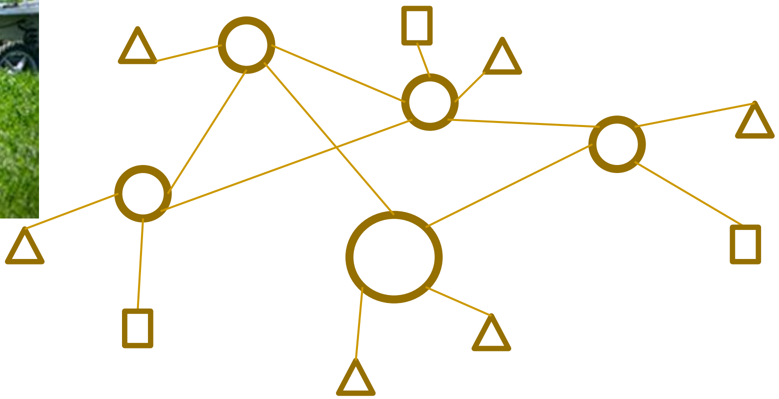
Validation of processing
- complex rules
- distributed reasoning

Scenarios : E-agriculture

On a real life experimental site in the Montoldre farm

Integration of sensors, mobile robots, weather conditions

- ❑ Irrigation
- ❑ Field access by machinery



Validation in presence of hazard
- Weather
- Open-field context

Long term experiments
- Energy consumption
- Time scaling

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

Some CoSWoT figures

4 laboratories,
1 company

1 M€
including 735k€
financed by ANR

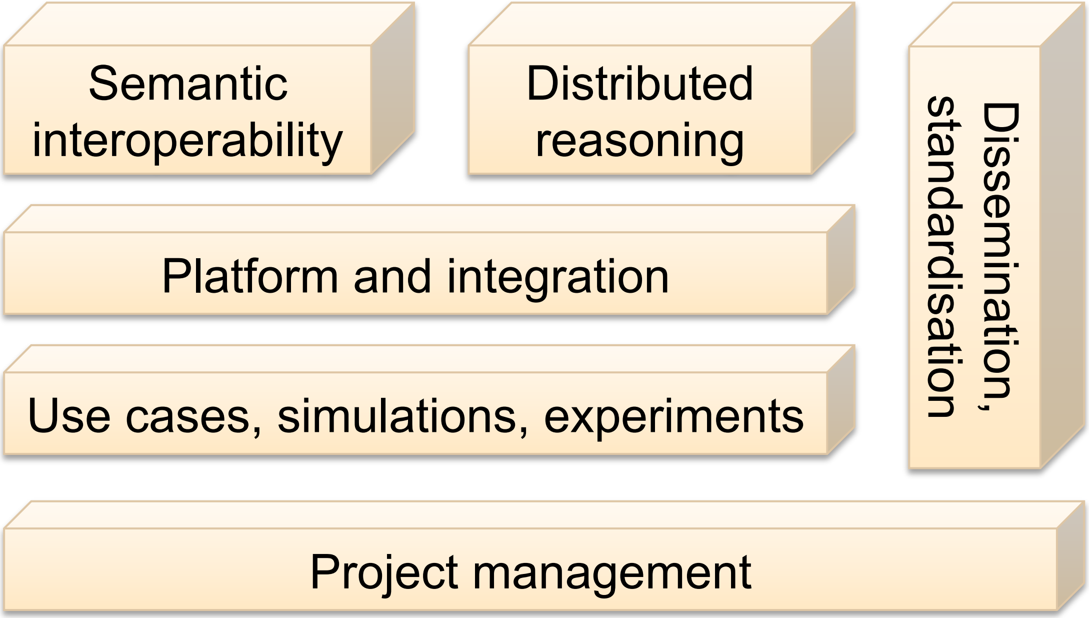
306 man*month:
17 researchers,
1 engineer,
2 post-docs,
2 co-supervised PhD students,
5 M2 internships

48 months
Starts Feb 18, 2020

2 application domains:
e-agriculture,
smart building

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

CoSWoT workpackages

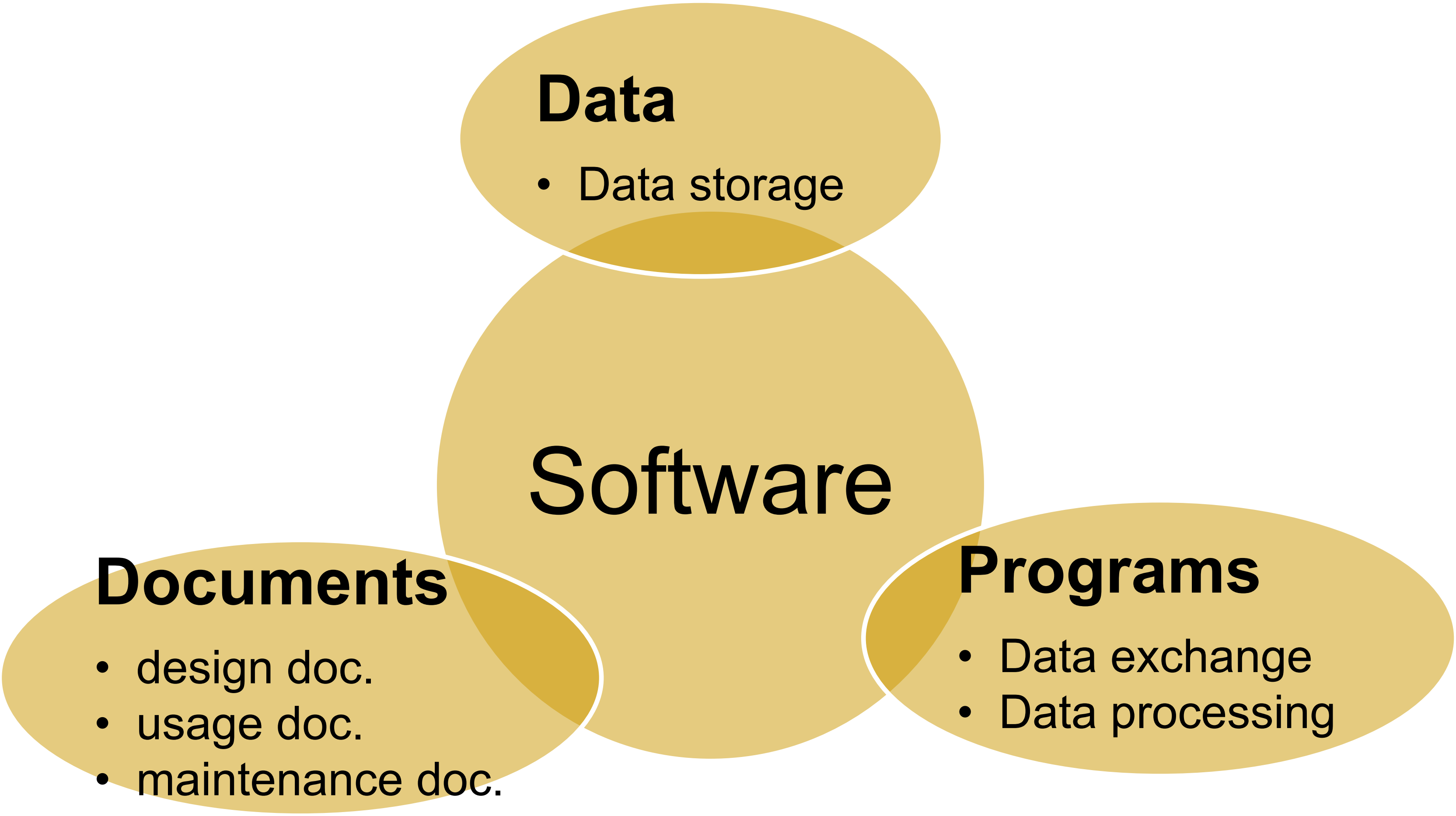

Semantic interoperability
Distributed reasoning
Platform and integration
Use cases, simulations, experiments
Dissemination, standardisation
Project management

ANR PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

Introduction

SOFTWARE ENGINEERING & UML MODELING

Definition : Software



Software

Data

- Data storage

Documents

- design doc.
- usage doc.
- maintenance doc.

Programs

- Data exchange
- Data processing

History : The 1968 software crisis

- Observations
  - Projects are late and budgets are exceeded
  - Processing errors are numerous
  - Systems are enormous
  - Developed softwares are not adopted by end users
- Software Engineering conference
  - The United Nations ask software specialists to attend
- Objectives
  - Improve software quality
  - Define software production methods

The software crisis remains

- From 1965 to 1995
  - Software volume * 100
  - Developers productivity * 3
- Microsoft Exchange Server development in1995
  - 1 000 person.year, 7 million lines
  - 30 lines per person per day
- Windows 2000 development
  - re-write 70% of the 16 million lines of Windows NT
  - 5 000 engineers, 3 years
  - 4.8 lines per person per day

Usual clients claims

- Specifications (needs) not respected
- Expected costs and dates overtaken
- Difficult maintenance
  - Bugs correction and versions evolution
- Performance under expectations
- Missing or unclear documentation
- Uncorrect reliability

Programming errors are not the only cause !!!

⇒ Software QUALITY, software engineering

Definition : Software engineering

- IEEE (ieee.org)

The application of a **systematic, disciplined, quantifiable** approach to the *development, operation, and maintenance* of software

Why is it so special to build software?

- Unique and without fatigue
  - ❑ Designed and built once, reproduction does not insert errors
  - ❑ Defects are not due to usage abuse but come from design
- Invisible
  - ❑ Software construction is a purely intellectual activity
  - ⇒ Quality is difficult to perceive
- Usually complex
  - ❑ Software are designed to help humans solve complex problems or tasks ; it is then complex by nature
- Methods look so constraining
  - ❑ It is easy to remain « artistic », heavy to follow a method

Some big bugs in history

- 14/07/2010 http://www.france.fr unavailable
  - Too many connections (25 000)
- 20/11/2008 http://www.voyages-sncf.com is KO
  - Too many bugs in the newly provided version
- 23/09/1999 Mars Climate Orbiter lost after a 9 months trip
  - Cause : confusion between feet and meters
    - Team work difficulty
  - Cost: 120 M$

Some big bugs in history

04/06/96, Explosion for the first launch of Ariane V

- Inertial platform software taken from Ariane IV without any new validation
- Ariane V has more powerful engines to cope with acceleration due to Earth rotation
- Sensors detected Ariane V angle, but the software judged it did not conform to the plan (of Ariane IV), and provoked self destruction

- 10 years, 38 billiard Francs, 39 seconds flight

Some big bugs in history

- 27/07/1962 Destruction of the Mariner1 space probe
  - 1 character error in a Fortran program
  - 80 millions dollars
- Crossing Equator line, a F16 get on its back
  - latitude sign change misunderstood
- 22/12/2001 750 000 CB payment terminals off
  - Saturation of payment authorization servers
    - Usually few tenths of seconds. 30 mn that day
  - Clients quit their full caddies in supermarkets. The Leclerc groups says they lost 2 million euros.

Software production = Team work

- Objectives : build a **product**
  - Software = program, data, documents
- Coordination of many **persons**
  - Clients, end users, development team…
- The job is organized in a **project**
  - Divide into activities organized following a workflow with a rigorous **planning**

Software engineering: skills and knowledge

- Operate a **method** to design, realize and maintain good quality software & Design and develop an application for information systems
  - Write a **technical specification document** for software
  - Know the different kinds of **software development processes**
- Design the **architecture** of an object oriented software
  - **Structure a software** with weakly coupled and highly cohesive packages and classes
- Use UML diagrams to **model** an object of study
  - Know the **different UML diagrams** and formalisms to design object oriented software
  - **Design, interpret, check** a set of UML diagrams modeling the same object of study
- Build **good** software
  - Operate advisedly **object oriented mechanisms** : inheritance, genericity, polymorphism...
- Operate a process to ensure and control software **quality**
  - Operate different types of **tests**
  - Set up and use tools for **collaborative work** (IDE, version management)

Software engineering - contents

- Introduction to software engineering Frédérique Laforest
- UML Frédérique Laforest
- Tests and Tools for software engineering Frédérique Laforest
- Requirement engineering Omar Hassan
- Software life cycle Omar Hassan
- Architectural patterns and security by design Omar Hassan
- Project during lab. sessions

Main reference : Ian Sommerville,*Software Engineering 9*, Addison-Wesley, 2010

http://www.cs.st-andrews.ac.uk/~ifs/Books/SE9/Presentations/index.html