Model-based Development for Embedded Control Systems

Paul Caspi

Laboratoire **Verimag** *(CNRS-UJF-INPG)*

⇒ **Which embedded control systems?**

⇒ **Aérospatiale pioneering role**

⇒ **State of the art**

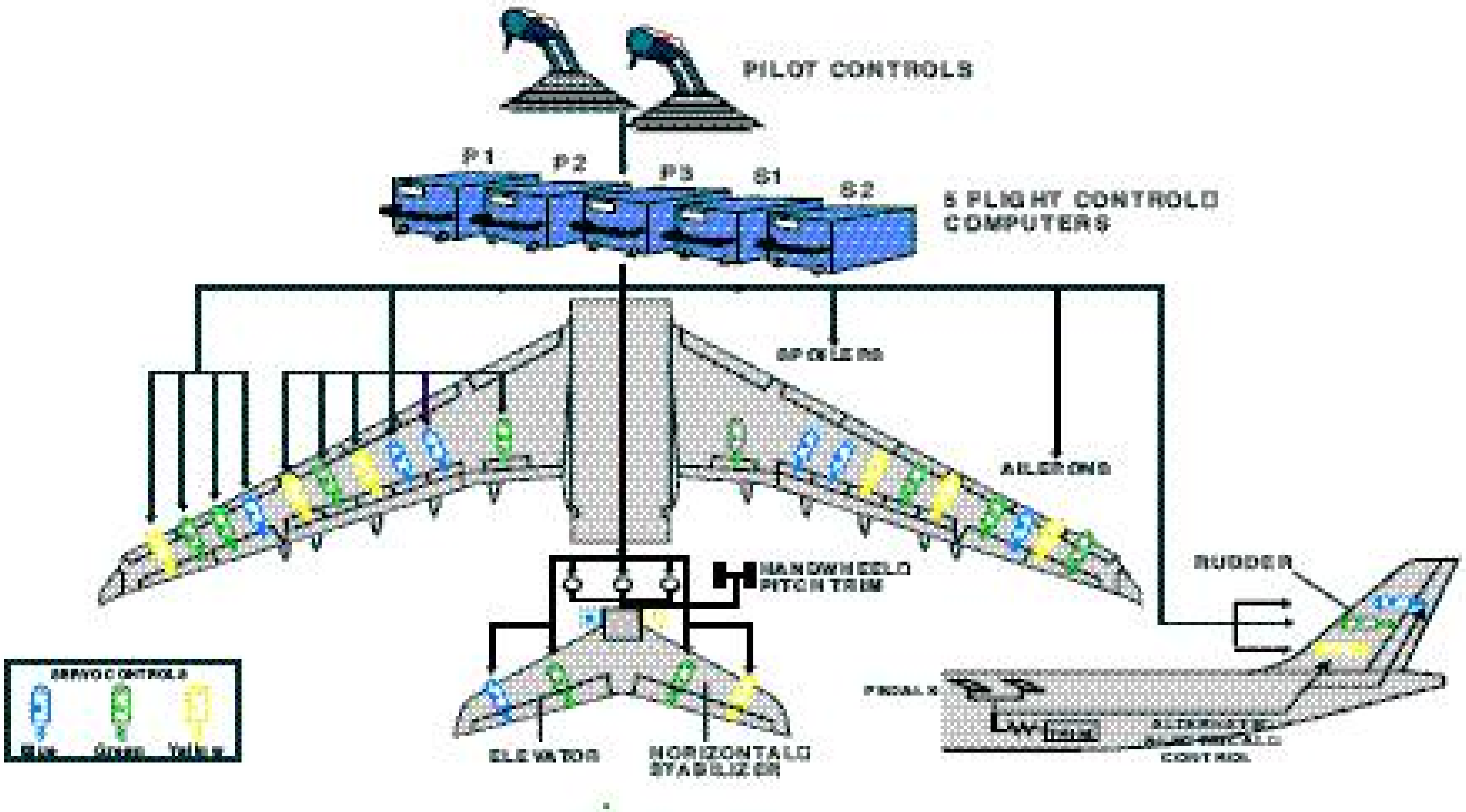
⇒ **Table of Contents**

Which Embedded Control Systems?

safety critical systems

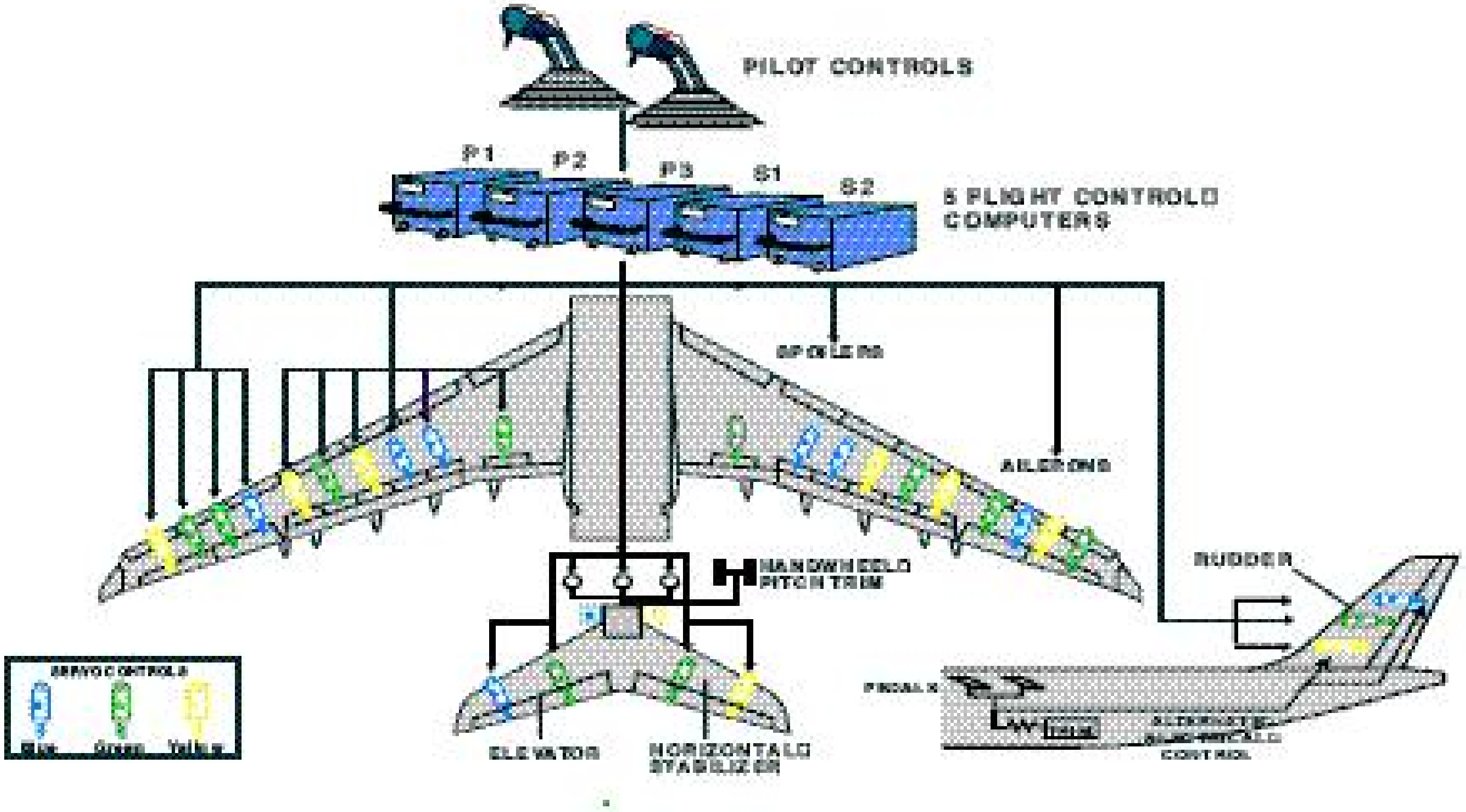
mission critical systems, time to market

Looking inside



Fly-by-wire ? Drive-by-wire ? Electronic Control Units ?

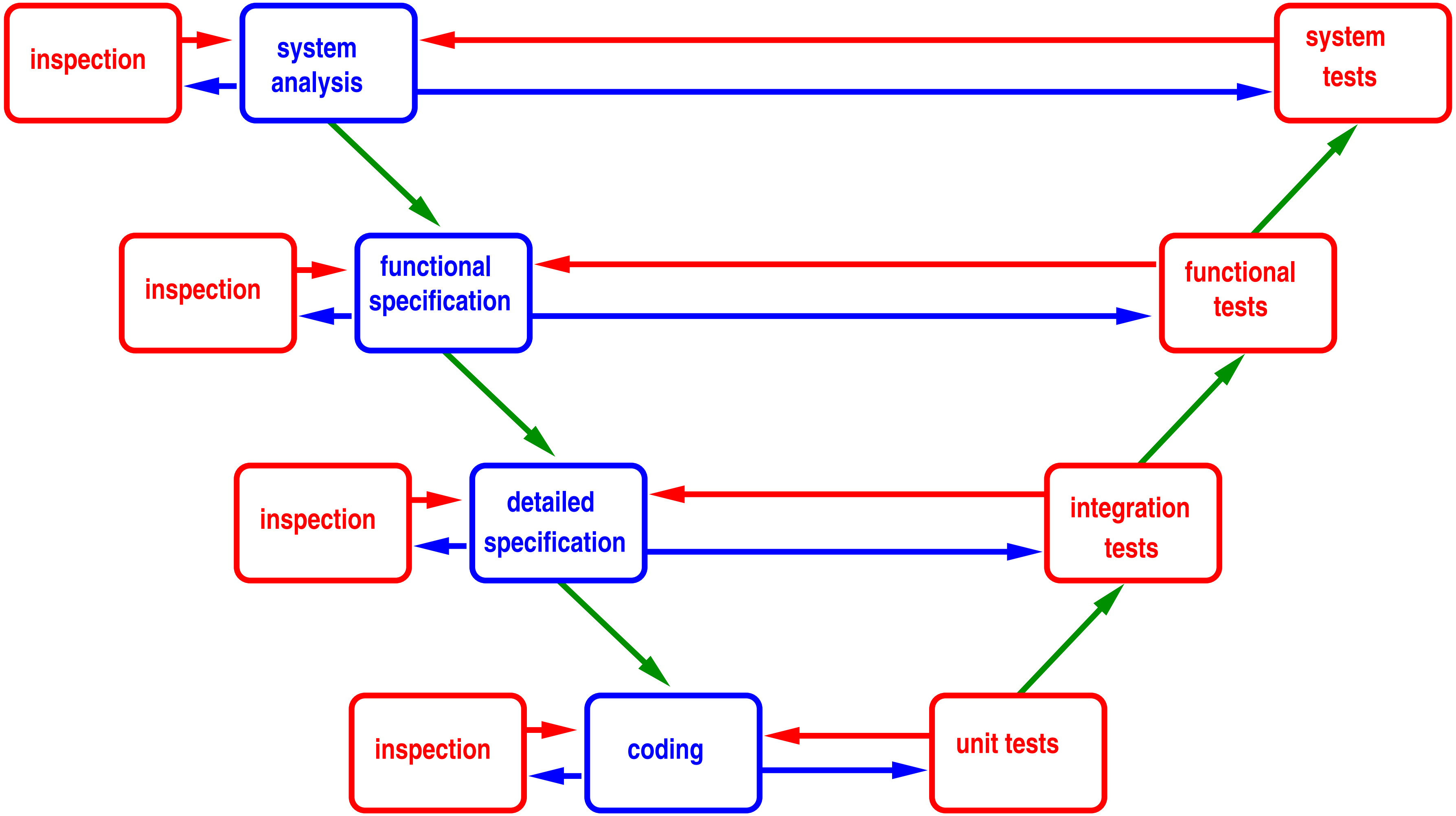
Looking inside



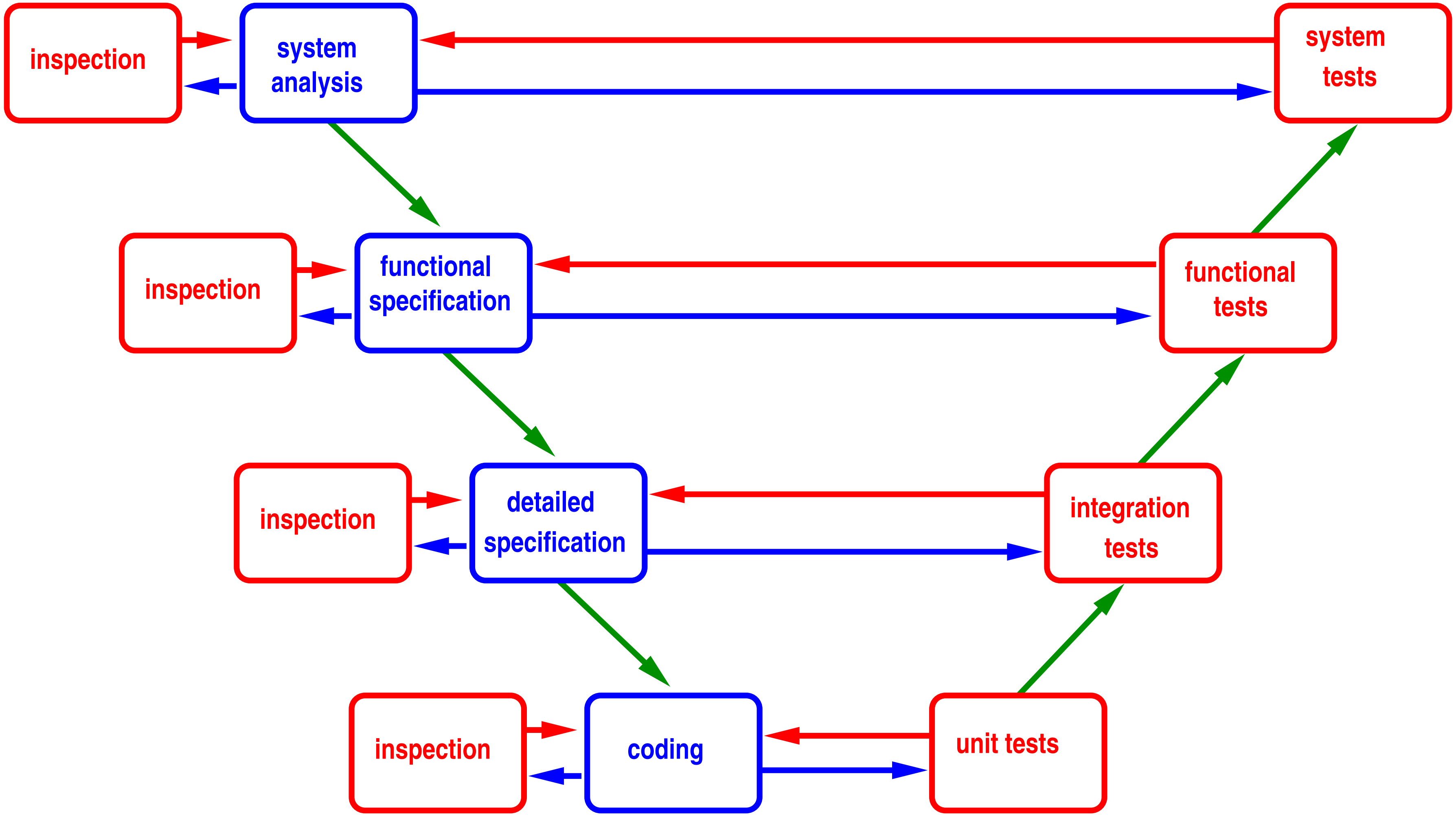
Fly-by-wire ? Drive-by-wire ? Electronic Control Units ?

Fly-by-computers ! Fly-by-software !

Traditional Ways to Critical Software/System



Traditional Ways to Critical Software/System



Is it the way we design bridges ? By trials and errors ?

Is it an engineering way ?

Model-based: move from this. . .

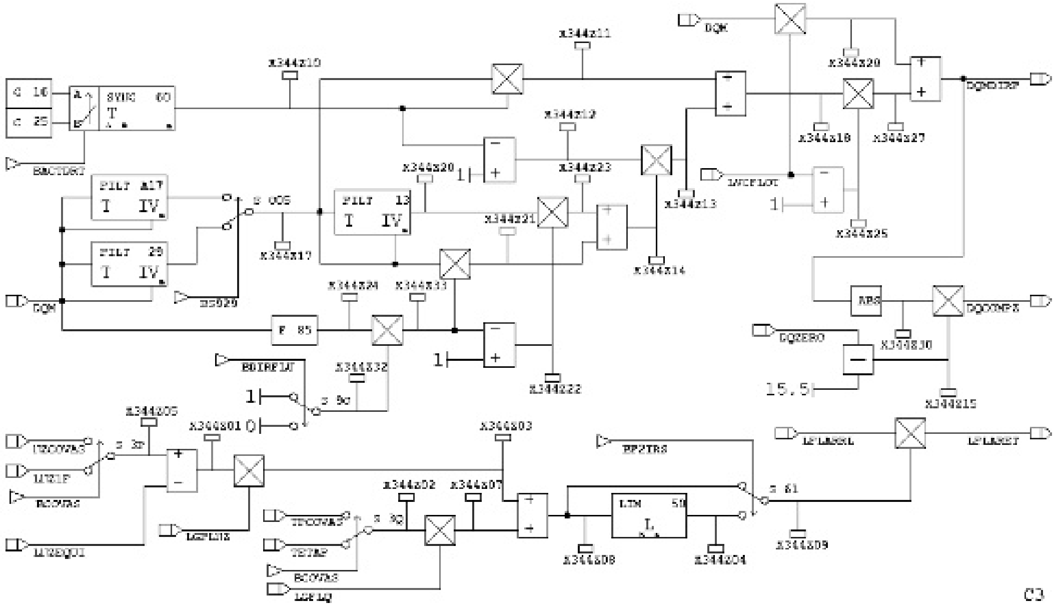
designed by trial and errors

to this. . .

model-based design

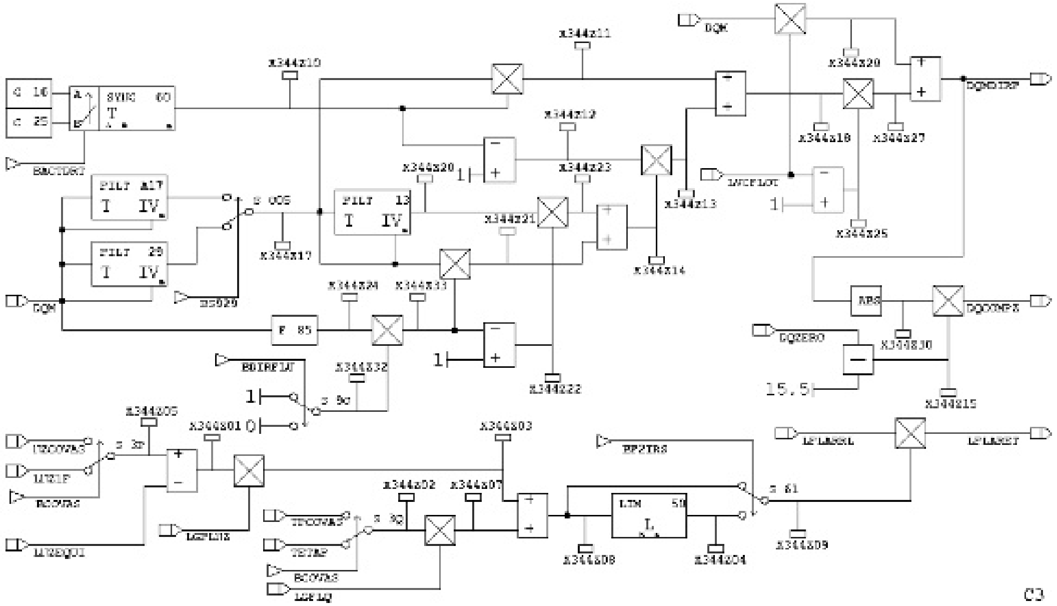
Aérospatiale pioneering steps in the early eighties

control models (block-diagrams)



Aérospatiale pioneering steps in the early eighties

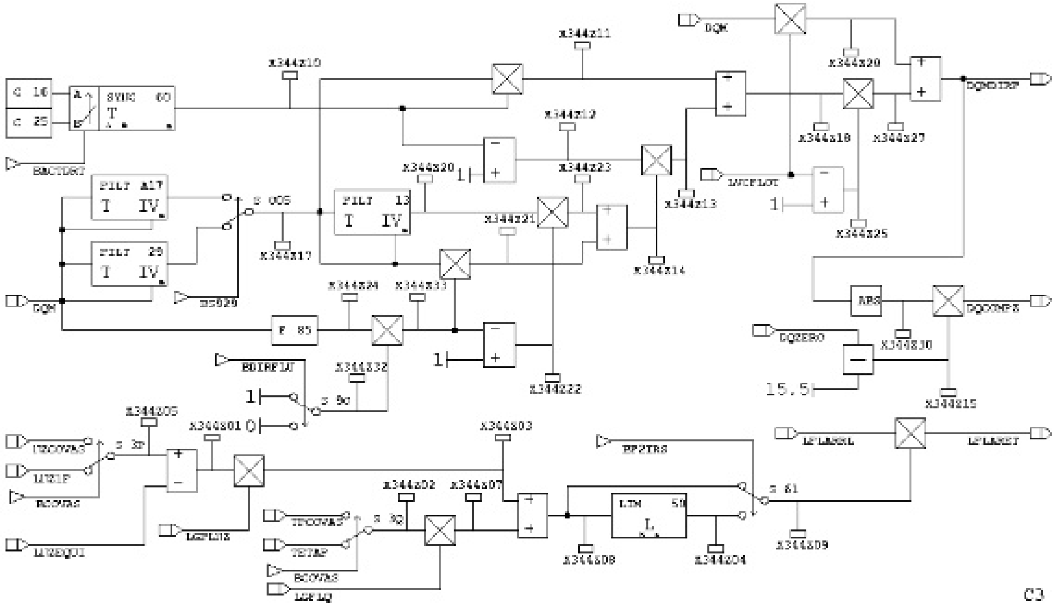
control models (block-diagrams)



= **formal software specification**

Aérospatiale pioneering steps in the early eighties

control models (block-diagrams)



= **formal software specification**

↓

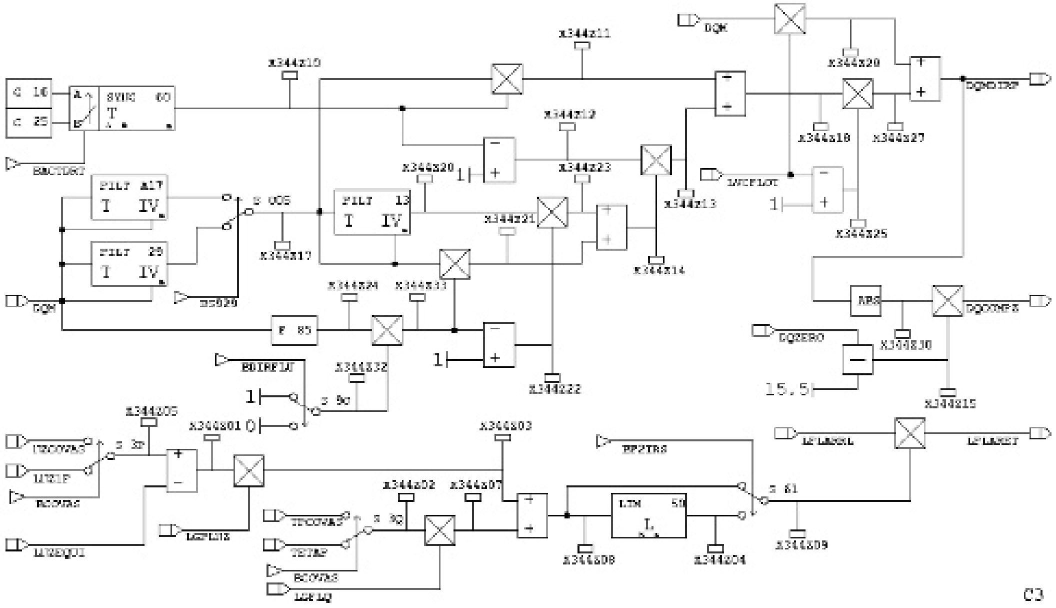
automatic code generation

↓

Software

Aérospatiale pioneering steps in the early eighties

control models (block-diagrams)



= formal software specification

↓

automatic code generation

↓

Software

“Spécification Assistée par Ordinateur”(SAO)

“Computer Aided Specification”

Interest of SAO

Twofold :

- **Automatic code generation from high-level control models:**

  easier and earlier debugging

- **Graphic language close to the cultural background of avionic engineers, test pilots, suppliers, certification authorities, . . . :**

  allows easier communication within the entreprise

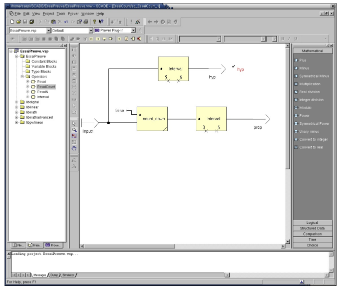
  preserves the know-how and makes easier the technology transfer

SAO participates to the success of A320

From then on. . .

Powerful model-based development tools:

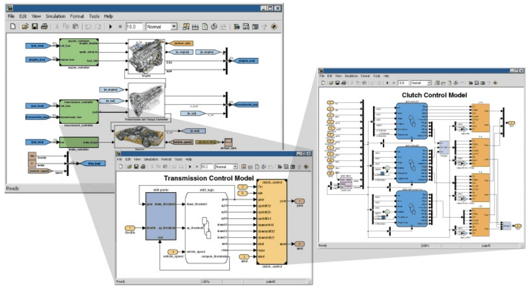
- SAO replaced by SCADE



commercial product partially based on synchronous technology

Do178B level A qualified automatic code generator

- Simulink/Stateflow



continuous/discrete time simulation toolbox

the defacto standard in control modelling

- Formal methods: automatic mathematical proofs for dynamic systems

PROVER TECHNOLOGY

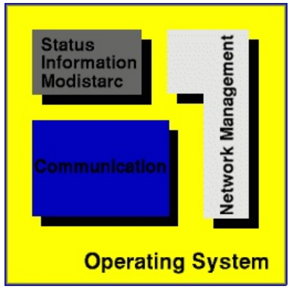
PolySpace TECHNOLOGIES Delivering safe code · faster

. . .

From then on. . .

More powerful execution platforms:

- multi-tasking



WIND RIVER

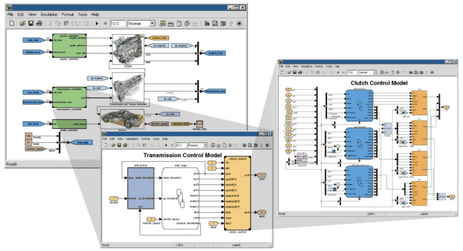
- distributed and multi-processor

TTTech

FlexRay™

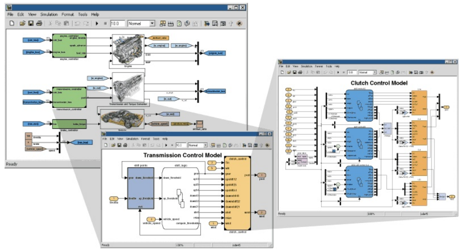
State of the Art

modelling

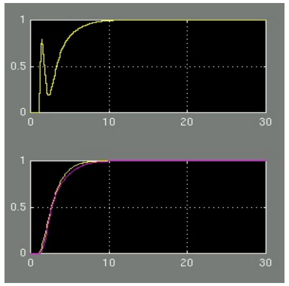


State of the Art

modelling

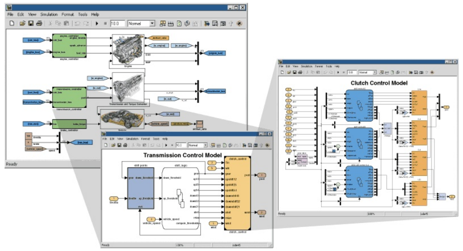


simulation
debugging

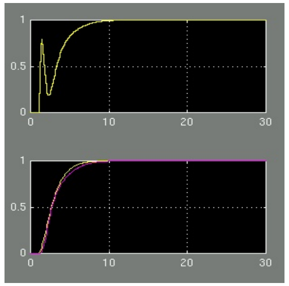


State of the Art

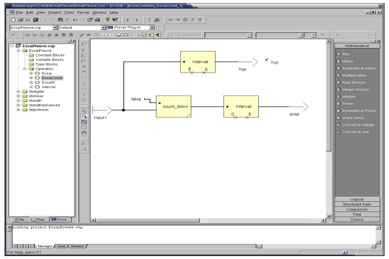
modelling



simulation
debugging

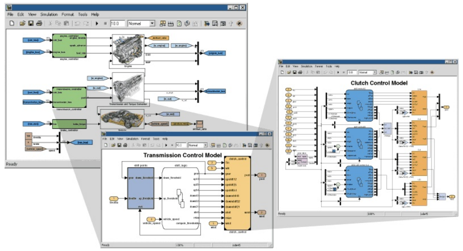


automatic import

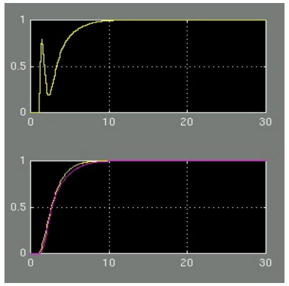


State of the Art

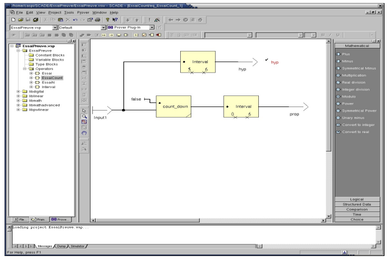
modelling



simulation
debugging



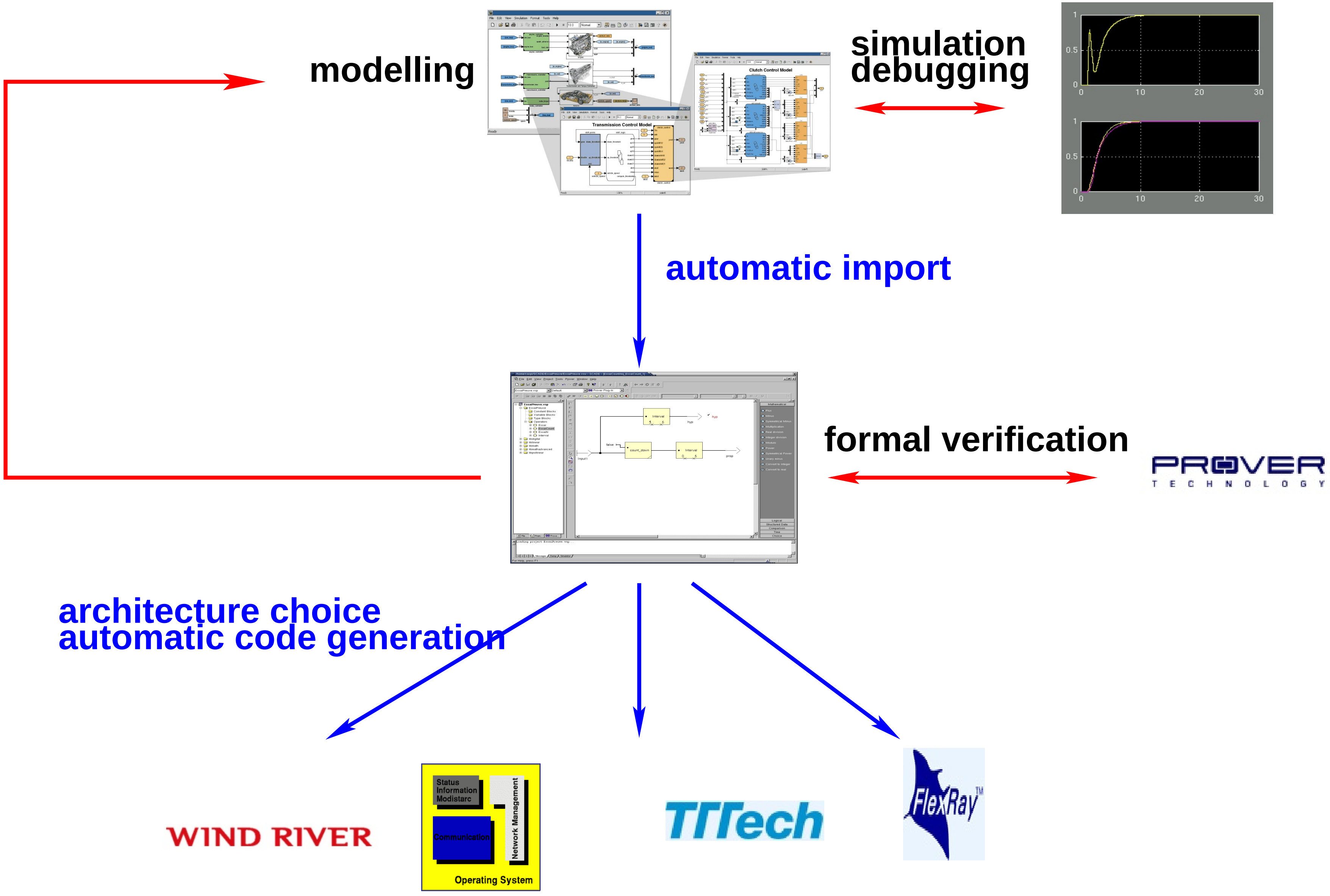
automatic import



formal verification

PROVER
TECHNOLOGY

State of the Art



modelling

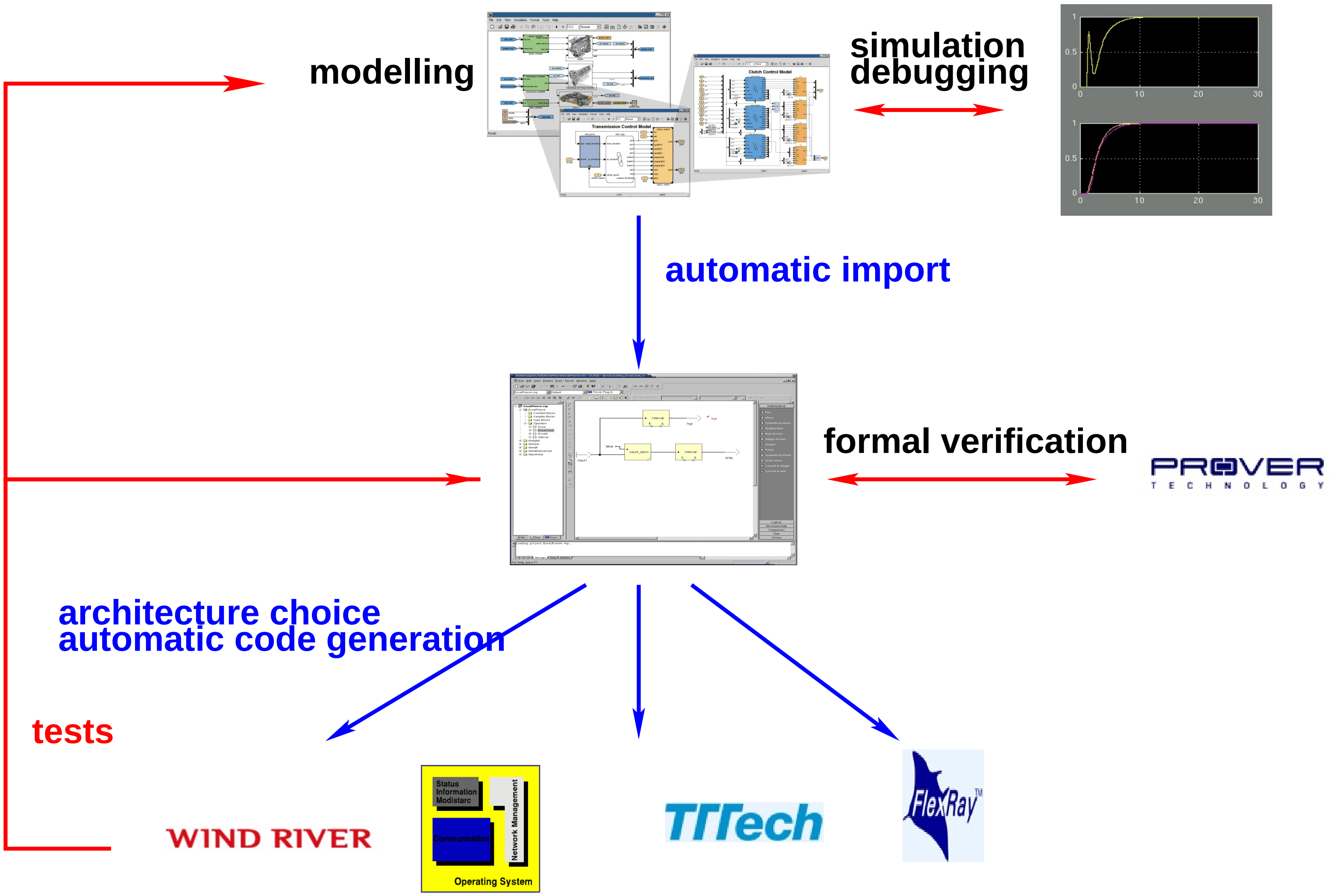
simulation
debugging

automatic import

formal verification

architecture choice
automatic code generation

State of the Art



modelling

simulation debugging

automatic import

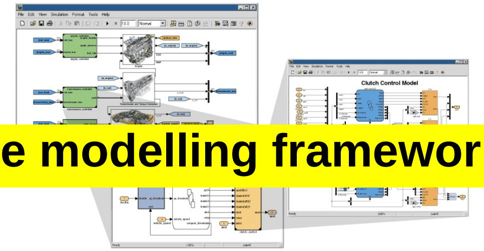
formal verification

architecture choice
automatic code generation

tests

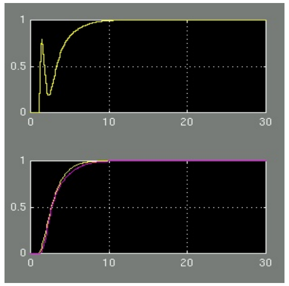
Perspectives

modelling

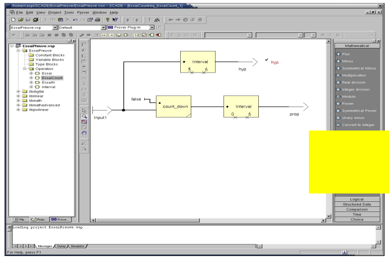


more modelling frameworks

simulation
debugging



automatic import



formal verification

PROVER
O L O G Y

more formal tools

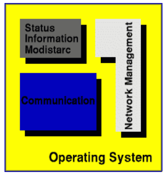
architecture choice
automatic code ger

more architectures

more test methods

tests

WIND RIVER



TTTech

FlexRay™

Perspectives

- more modelling frameworks:

  networks, telecommunications, . . .

- more powerful formal methods

- more execution platforms

  CAN, Ethernet, Internet, . . .

- more test methods

A Key Issue: Faithfulness

$$\boxed{\textbf{What you} \begin{cases} model \\ simulate \\ prove \end{cases} \textbf{is what you} \begin{cases} implement \\ execute \end{cases}}$$

(Gérard Berry 1984)

What does it mean?

Outline of the Course

- Simulink
- Stateflow
- Code generation
- Multi-threading
- Faithfulness