Model-based development for embedded control systems

Paul Caspi

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

28th September 2007

Model-based development for embedded control systems

Paul Caspi

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

28th September 2007

Many thanks to

- the organisers, speakers, attendants

for this wonderful and memorable day

Joint work with. . .

Jacques Richalet
Catherine Bellon
Eric Pilaud
Paul Amblard
Albert Benveniste
Benoît Caillaud
René David
Oded Maler
Thierry Le Sergent
Norman Scaife
Alberto Sangiovanni-Vincentelli
Gabriele Saucier
Nicolas Halbwachs
Daniel Pilaud
Joseph Sifakis
Pascal Raymond
Christine Bodennec
Moez Yeddes
Florence Maraninchi
Stavros Tripakis
Chiheb Kossentini
Anne Guérin-Dugué
Jacques Pulou
John Plaice
Jean-Louis Bergerand
Gérard Berry
Alain Girault
Jean-Louis Camus
Marc Pouzet
Cécile Dumas
Adrian Curic
Christos Sofronis
Thao Dang

Joint work with. . .

Jacques Richalet
Catherine Bellon
Eric Pilaud
Paul Amblard
Albert Benveniste
Benoît Caillaud
René David
Oded Maler
Thierry Le Sergent
Norman Scaife
Alberto Sangiovanni-Vincentelli
Gabriele Saucier
Nicolas Halbwachs
Daniel Pilaud
Joseph Sifakis
Pascal Raymond
Christine Bodennec
Moez Yeddes
Florence Maraninchi
Stavros Tripakis
Chiheb Kossentini
Anne Guérin-Dugué
Jacques Pulou
John Plaice
Jean-Louis Bergerand
Gérard Berry
Alain Girault
Jean-Louis Camus
Marc Pouzet
Cécile Dumas
Adrian Curic
Christos Sofronis
Thao Dang

. . . and all those who will hate me for omitting to cite their name

Which Embedded Control Systems?

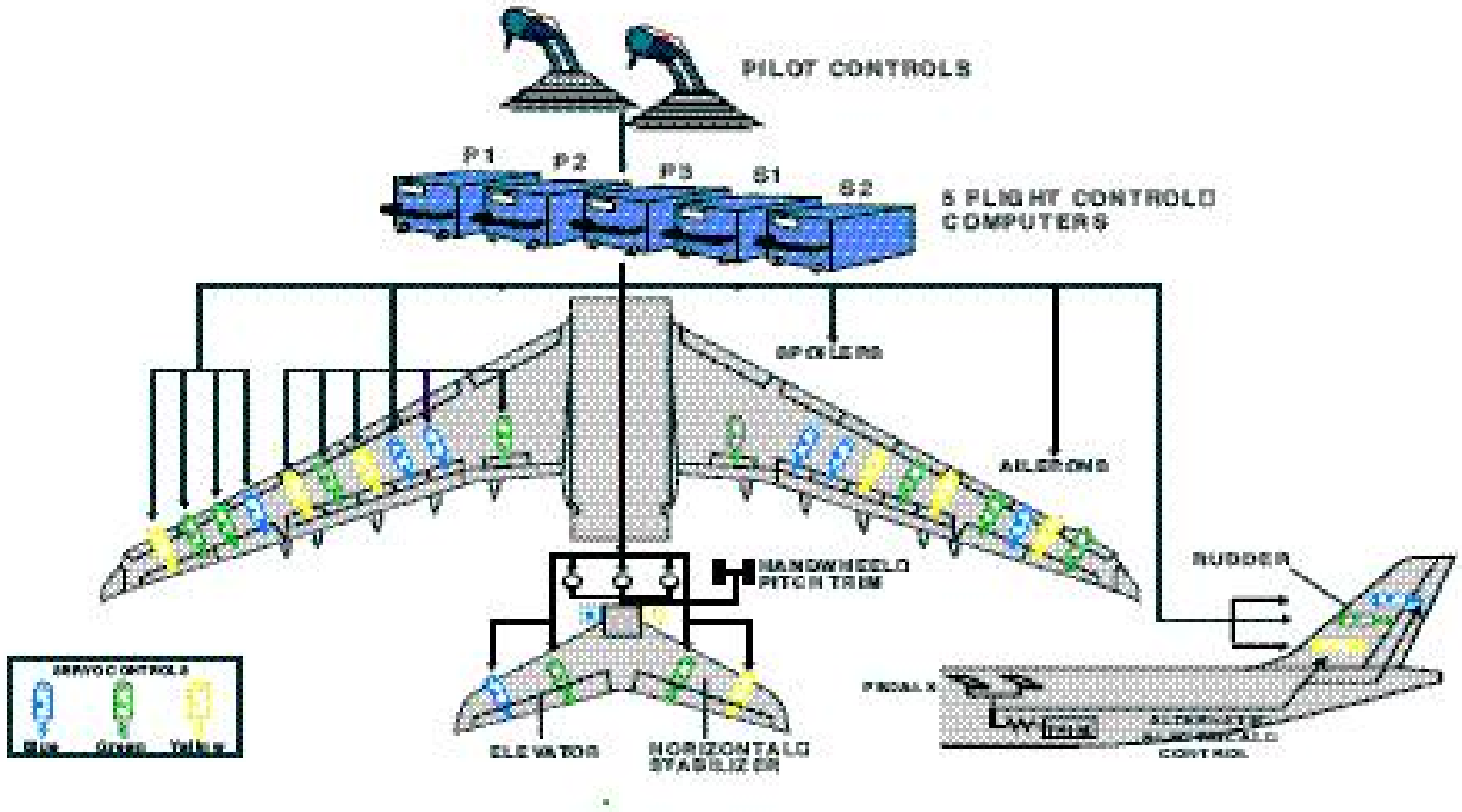
flight control

emergency shutdown

speed control, signalling

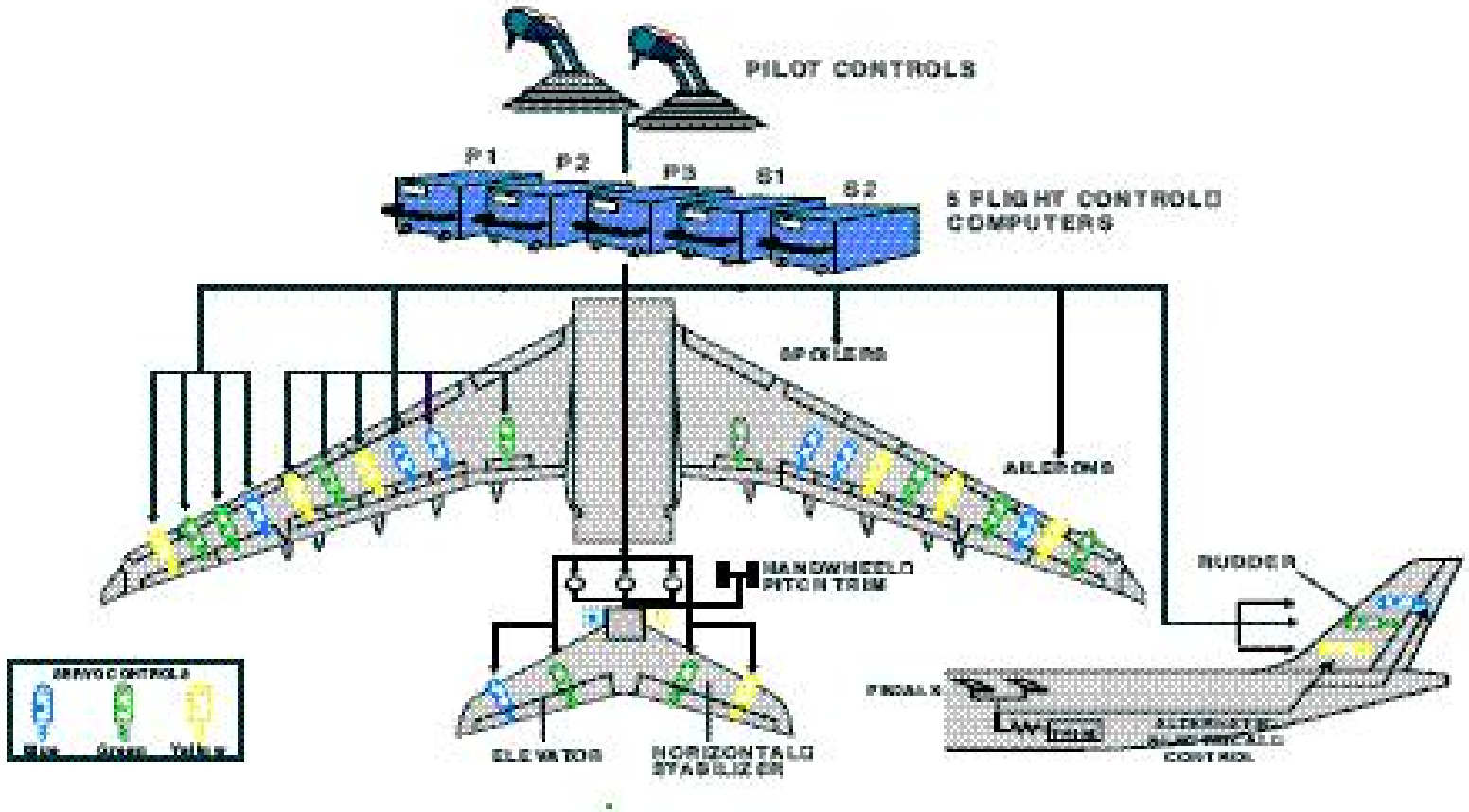
full automation

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 !

Two Questions

Knowing the low reliability of computing technology

- thousands of car “recalled” for computing bugs
- Ariane V accident
- your personal computer . . .

Two Questions

Knowing the low reliability of computing technology

- thousands of car “recalled” for computing bugs
- Ariane V accident
- your personal computer . . .

1. *Is it wise to use this poor technology in safety critical systems?*

Two Questions

Knowing the low reliability of computing technology

- thousands of car “recalled” for computing bugs
- Ariane V accident
- your personal computer . . .

1. *Is it wise to use this poor technology in safety critical systems?*

2. *Why, nevertheless, things are not as bad as could be expected?*

A Tentative Answer

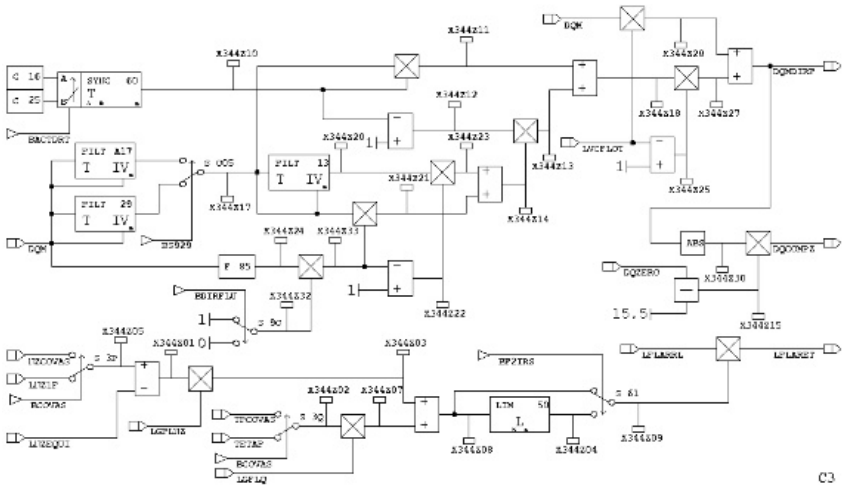
The safety-critical control industry has designed a very strong model-based development method
A short story of this method:

- **Aérospatiale** pioneering role
- How things evolved since then
- State of the Art and perspectives

Are academic people really aware of this story?

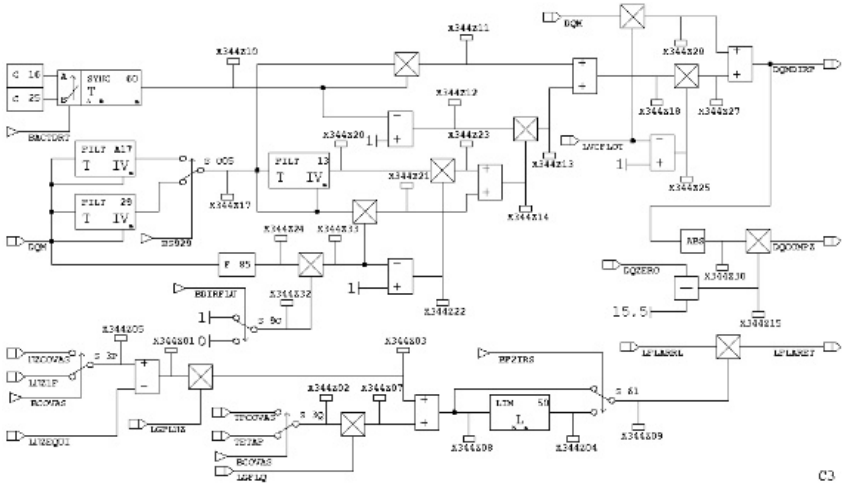
Aérospatiale pioneering steps in the early eighties

control models (block-diagrams)



Aérospatiale pioneering steps in the early eighties

control models (block-diagrams)

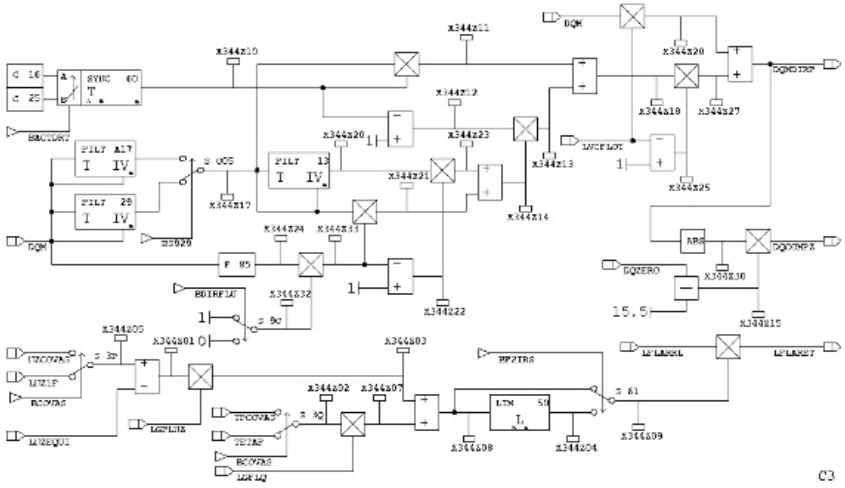


=

formal software specification

Aérospatiale pioneering steps in the early eighties

control models (block-diagrams)



↓

automatic code generation

↓

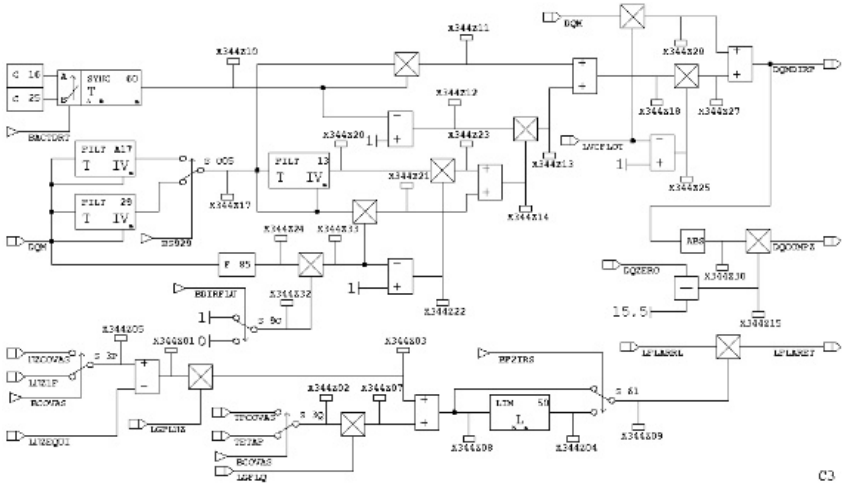
Software

=

formal software specification

Aérospatiale pioneering steps in the early eighties

control models (block-diagrams)



↓

automatic code generation

↓

Software

=

formal software specification

“Spécification Assistée par Ordinateur”(SAO)
“Computer Aided Specification”

Interest of **SAO**

Twofold :

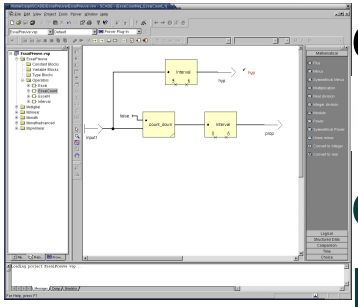
- Automatic code generation from high-level control models:
  easier and earlier debugging
- Graphic language close to the cultural background of avionic engineers, test pilots, suppliers, certification authorities, . . . :
  allows easier communication within the entreprise
  preserves the know-how and makes easier the technology transfer

SAO participates to the success of A320

From then on. . .

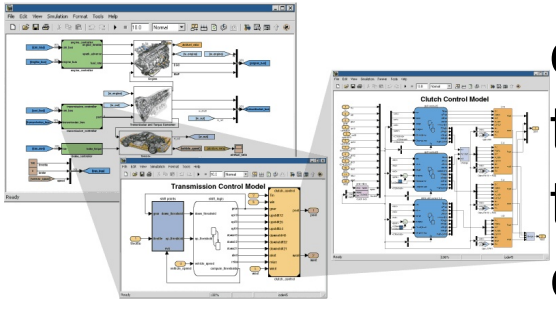
Powerful model-based development tools:

- **SAO** replaced by **SCADE**



commercial product partially based on synchronous technology
qualified code generator for safety-critical applications

- Simulink/Stateflow



continuous/discrete time simulation toolbox
the defacto standard in control modelling

- Formal methods: automatic mathematical proofs for dynamic systems

PROVER TECHNOLOGY

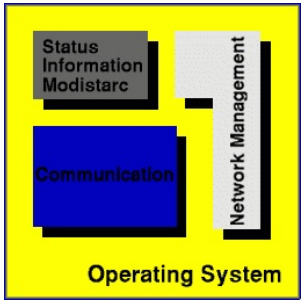
PolySpace TECHNOLOGIES
Delivering safe code · faster

. . .

From then on. . .

More powerful execution platforms:

- multi-tasking



WIND RIVER

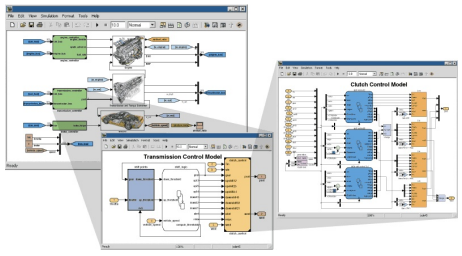
- distributed and multi-processor

TTTech

FlexRay™

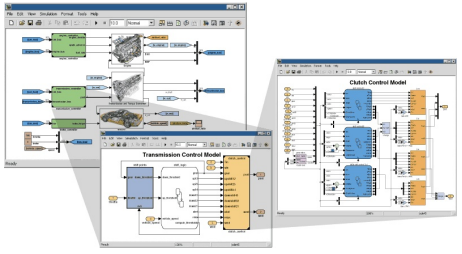
State of the Art

modelling

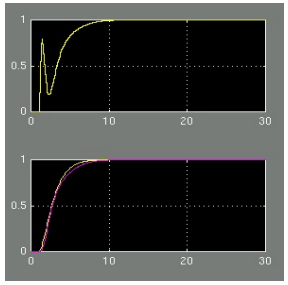


State of the Art

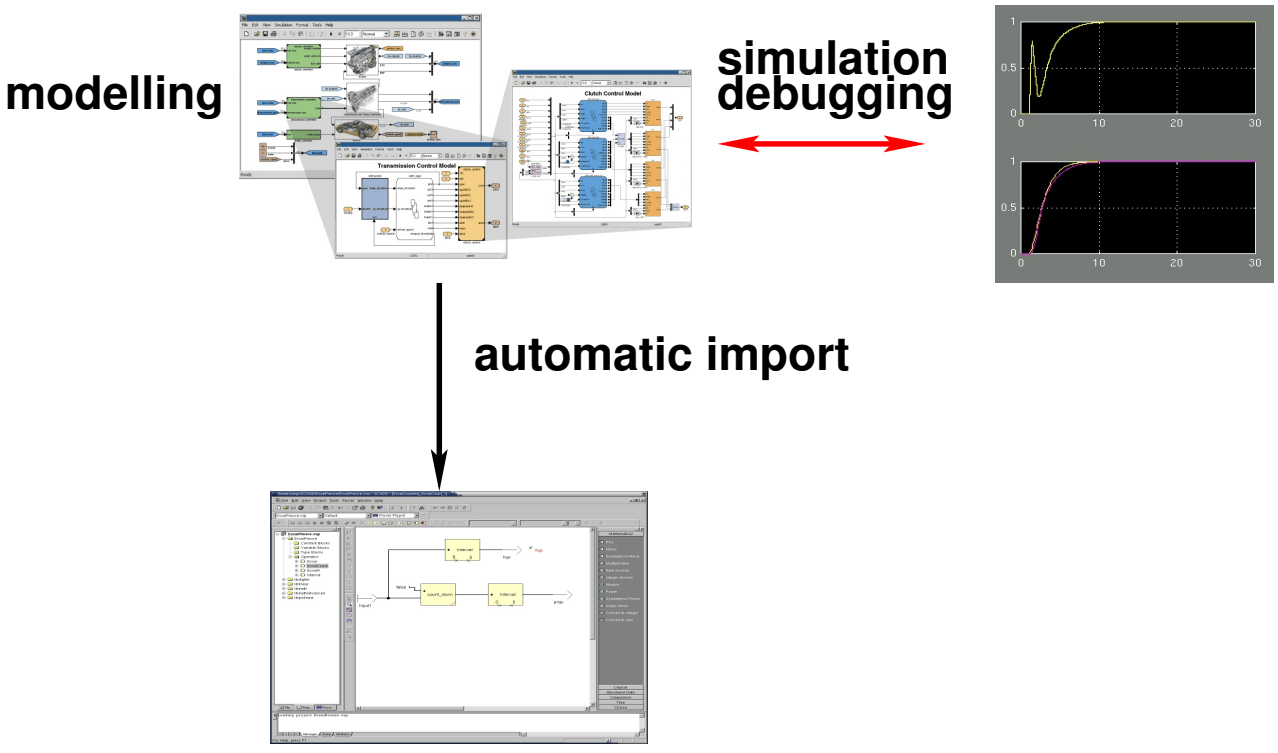
modelling



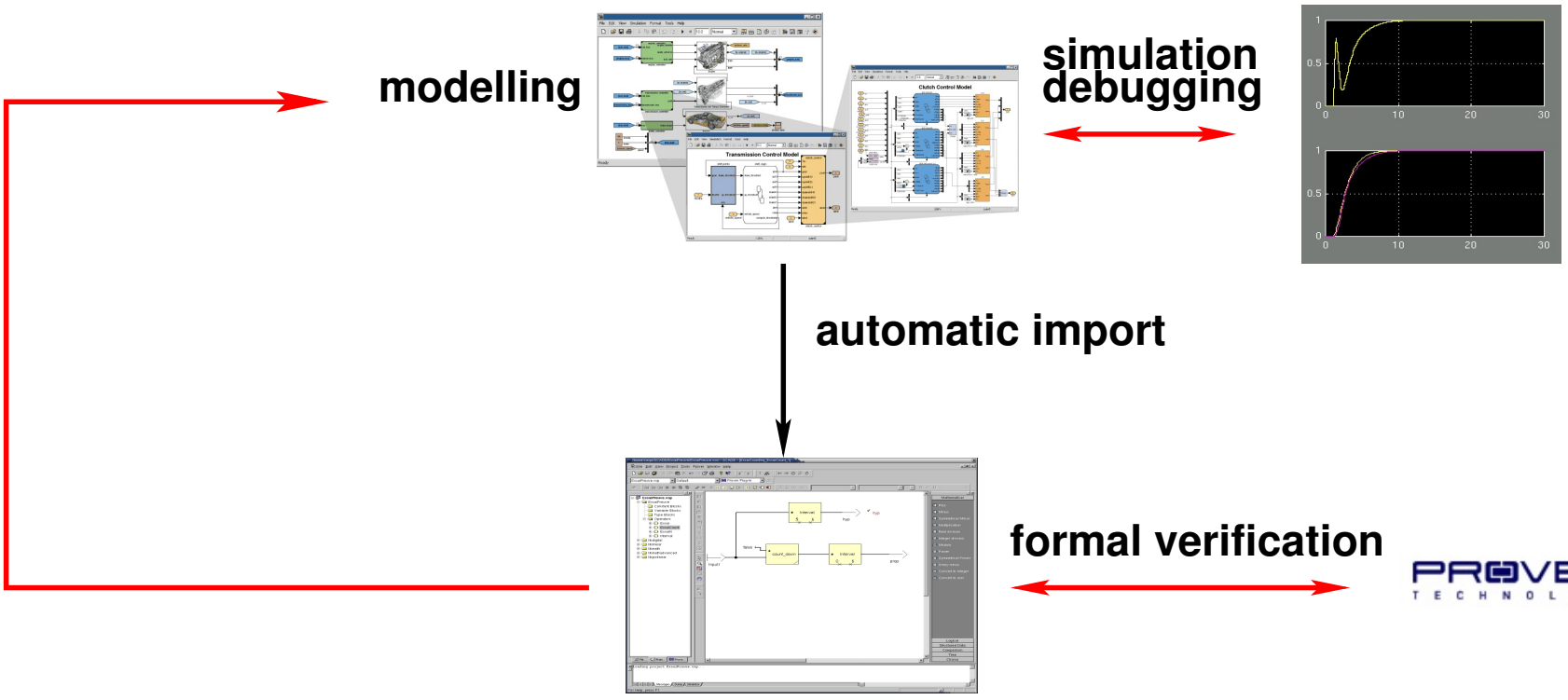
simulation
debugging



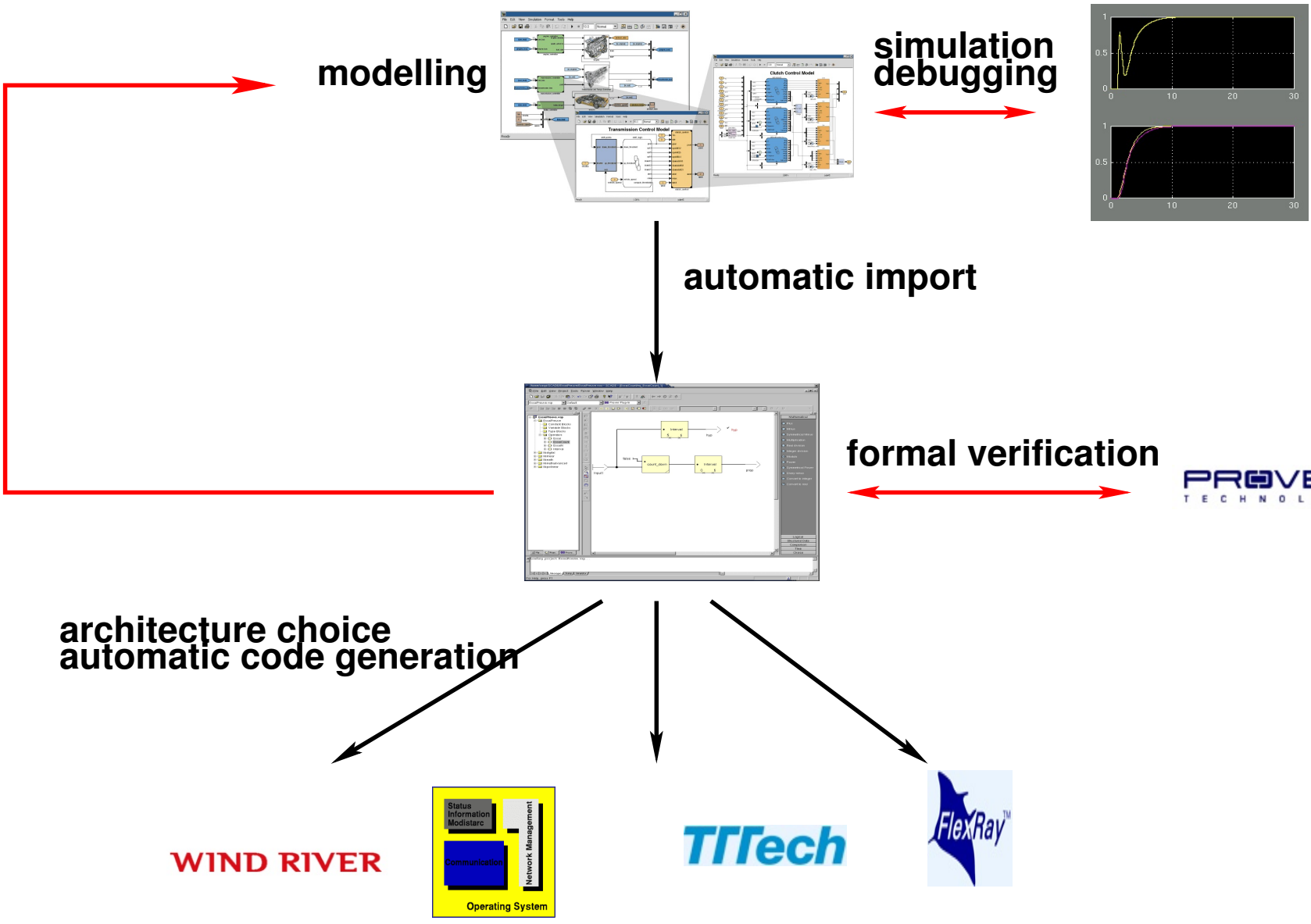
State of the Art



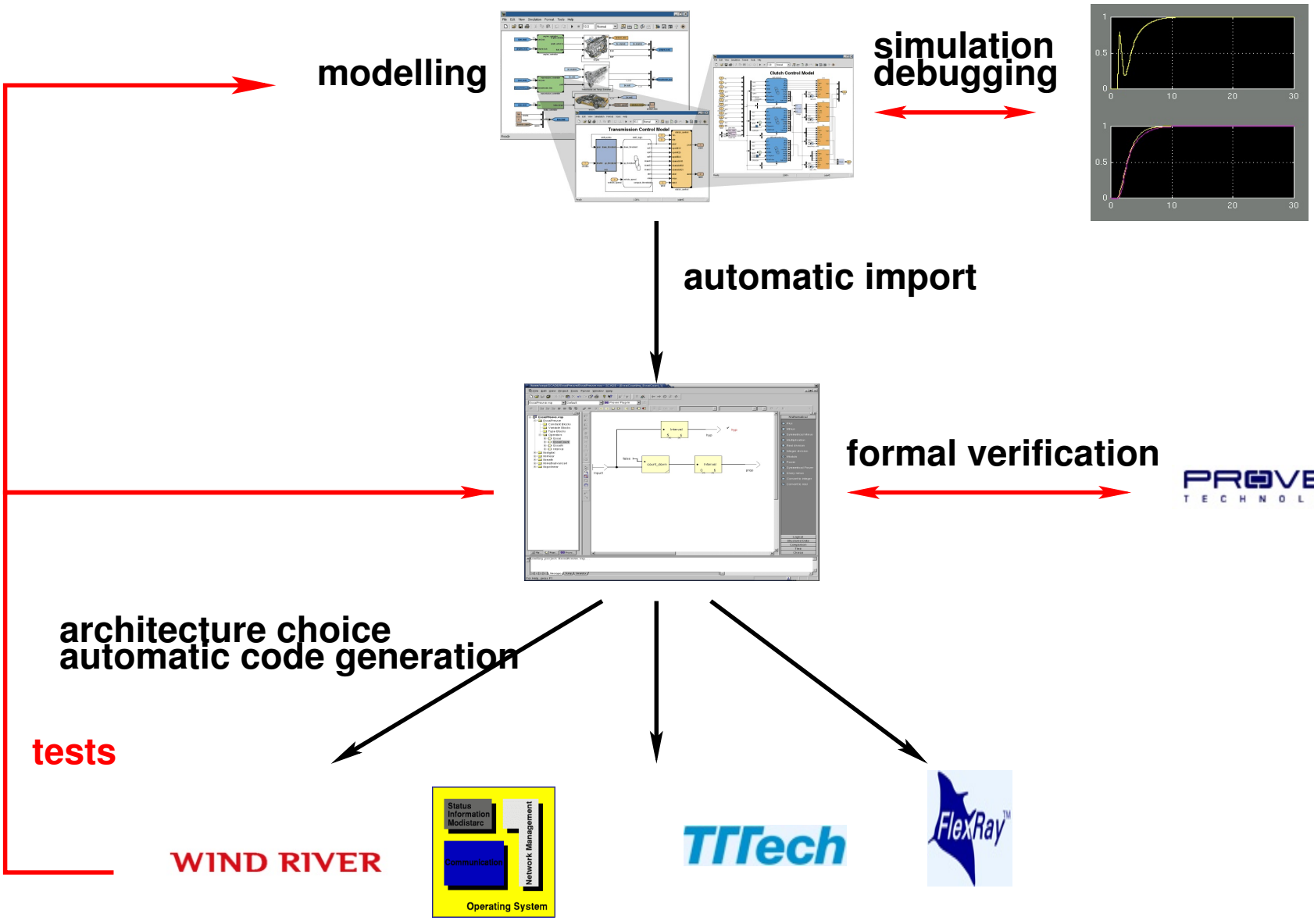
State of the Art



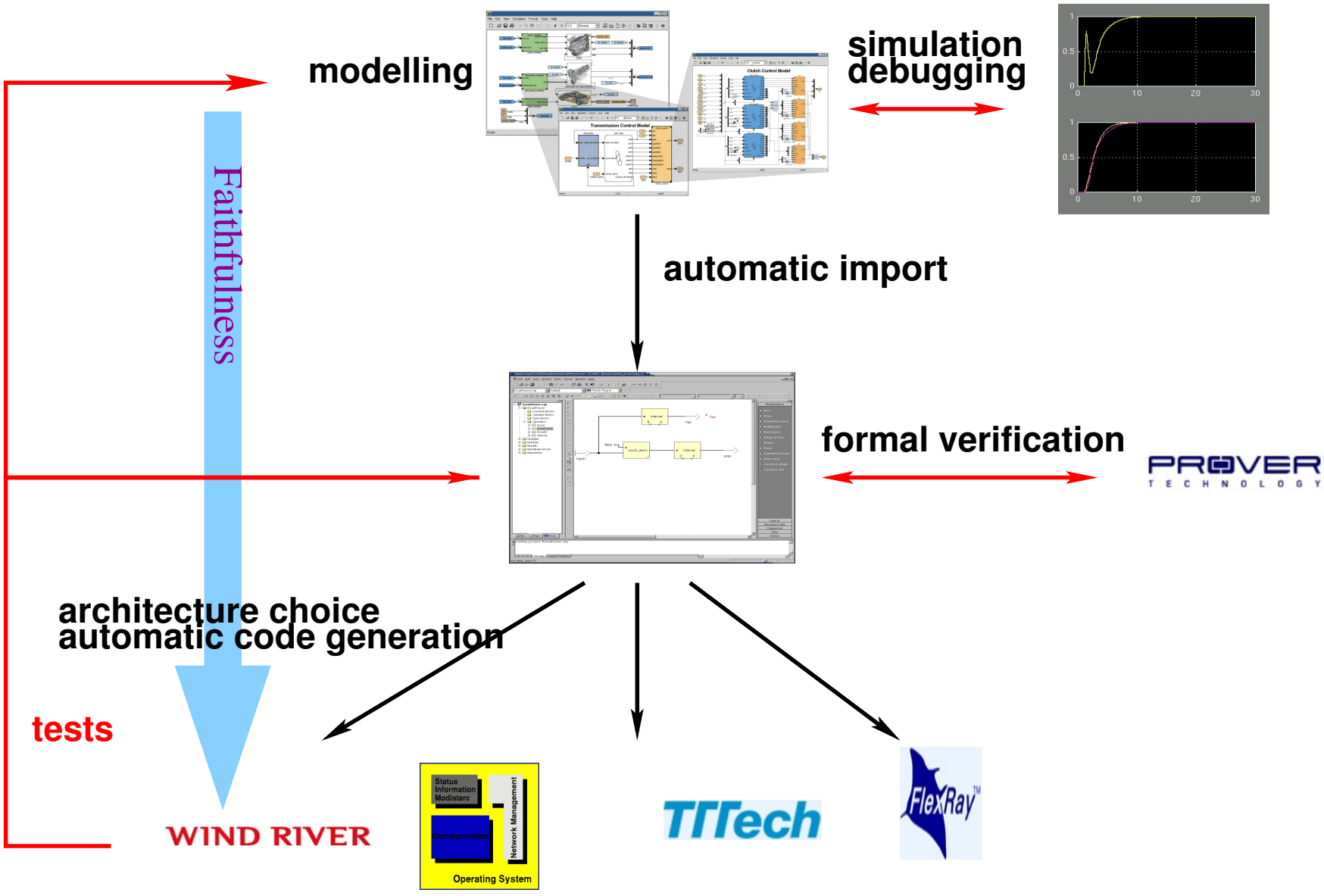
State of the Art



State of the Art



State of the Art



A Key Issue: Faithfulness

What you { *model* / *simulate* / *prove* } is what you { *implement* / *execute* }

(Gérard Berry 1984)

From Handicraft to Industry

In twenty years, the industry of critical control moved from

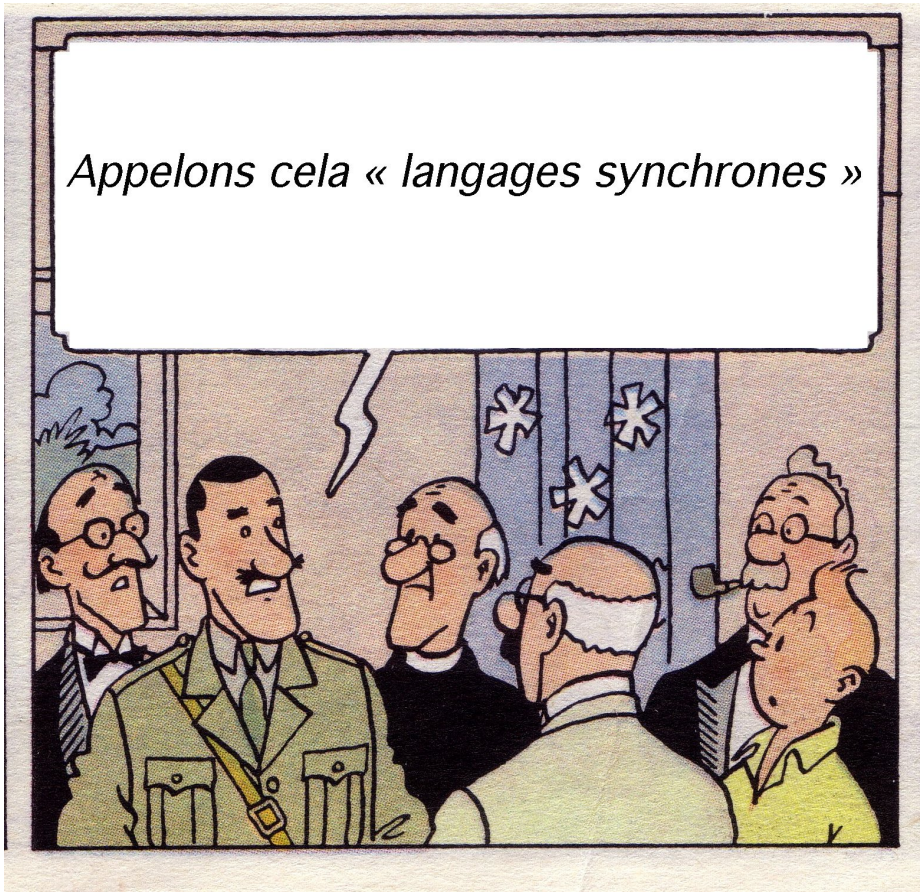
- handicraft :
  - paper design, human coding, validation on hardware
- to industry:
  - functional and architectural design and validation based on formal models that can be simulated and checked,
  - automatic code generation ensuring faithfulness between models and implementations

This is a notable advance that has to be pursued, strengthened and extended

In twenty years, we moved from this. . .

to this. . .

What was our role in this story?



Strengthen

Formalise

Generalise

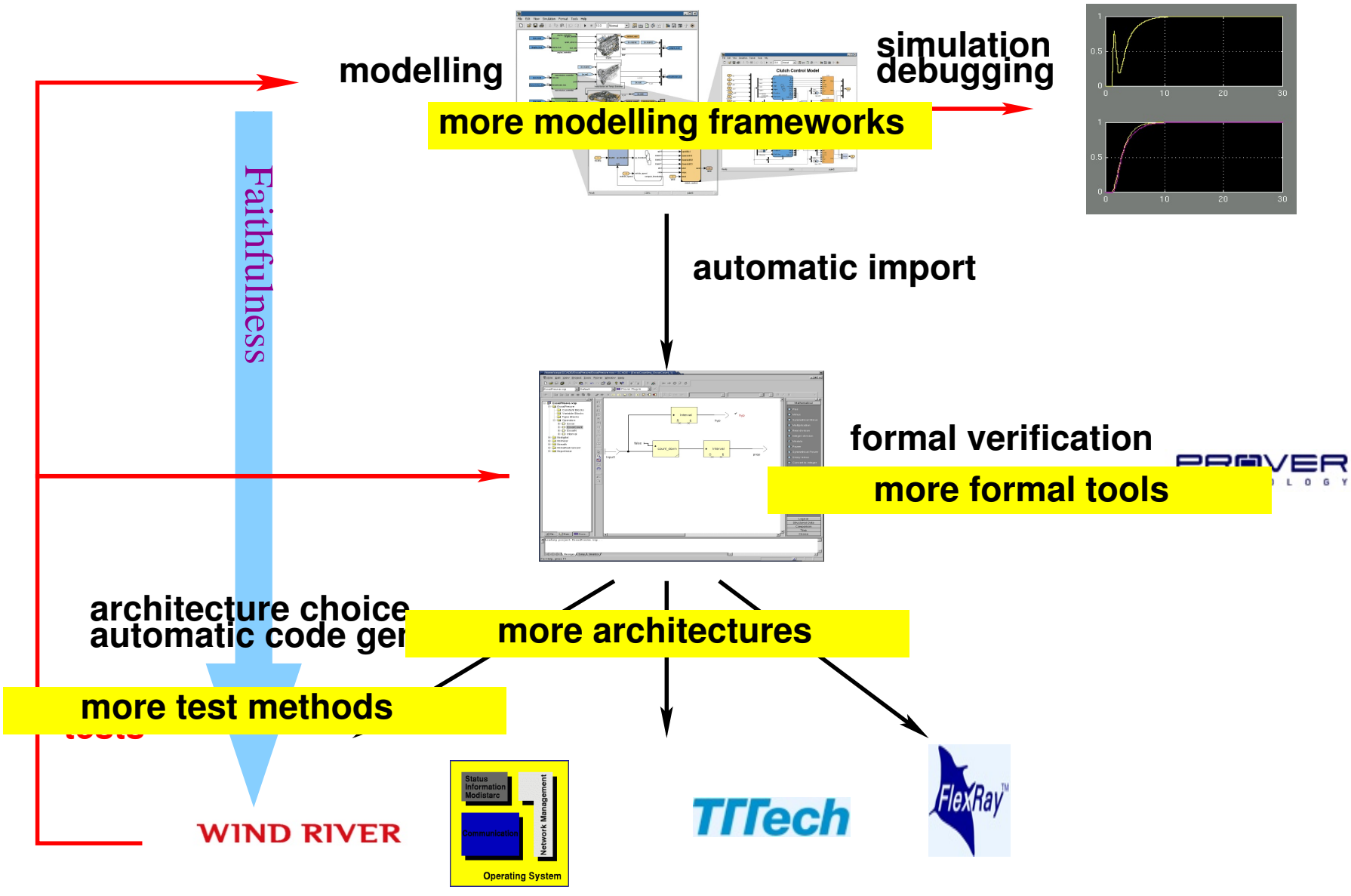
Optimise

Help building tools

Some Regrets

- We were ahead of Mathworks in the eighties and we missed continuous time
- I was a bad supervisor: as I didn't care of my career, I couldn't as well care for my students ones.

Perspectives



Thanks again to all of you and good luck

The True Story of SCADE/Lustre