A Formal Model-Based Tool for Reliable Interactive Prototyping of Safety Critical Highly Interactive Applications

Philippe Palanque

UNIVERSITE PAUL SABATIER TOULOUSE III

LIIHS-IRIT, University of Toulouse 3

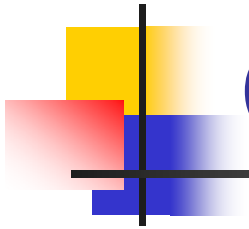
France

LIIHS

Safety Critical Interactive Systems

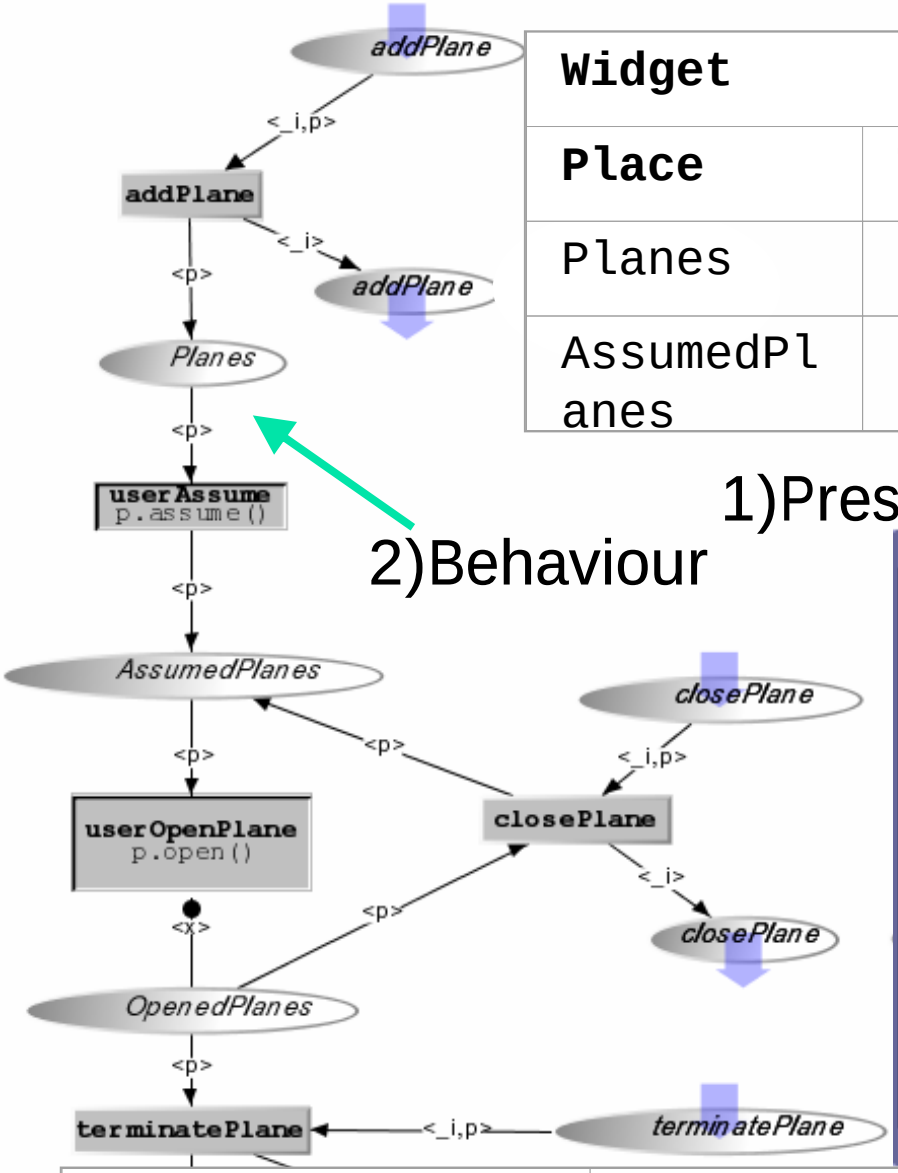
- Safety Critical Systems
  - Software Engineers
  - System centered
  - Reliability
  - [illegible]
  - [illegible]
  - Verification / Proof
  - Waterfall model / structured
- Interactive Systems
  - Usability experts
  - User centered
  - Usability
  - [illegible]
  - [illegible]
  - Evaluation
  - Iterative process / Prototyping
  - Novel Interaction techniques

Reliability Usability



Objectives

- To bring closer Usability and Reliability factors
  - How to include HF into software development process
  - How to build reliable Interactive Systems
- To ease and support communication between the various actors through models
- To provide techniques for assessing safety-critical interactive systems
  - By verifying properties
  - By assessing the complexity of the system (performance prediction)

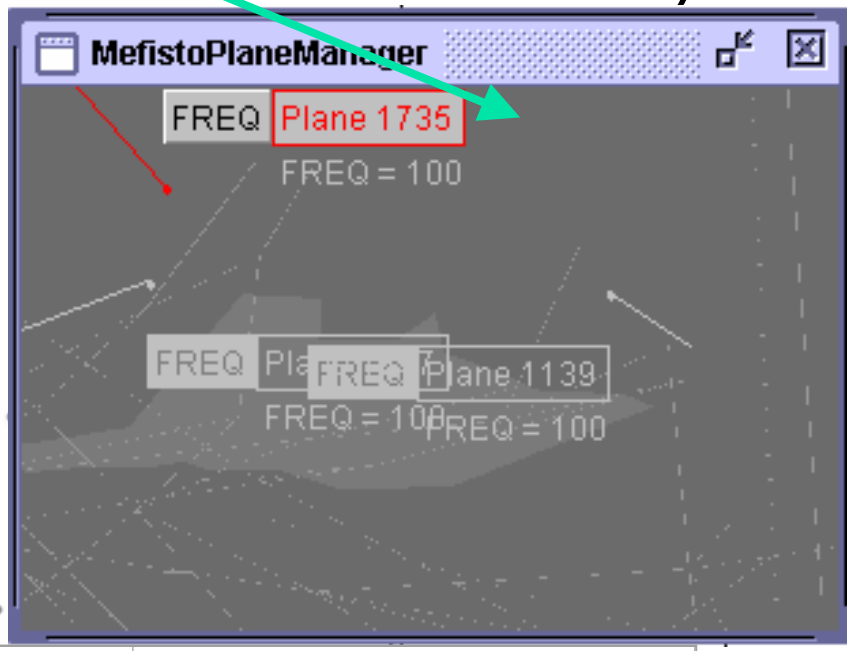


Widget		Event	Service
Place	Type		
Planes	Plane	LabeClick	userAssume
AssumedPlanes	Plane	ButtonClick	userOpenMenu

1)Presentation

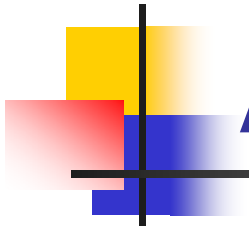
2)Behaviour

3)Activation

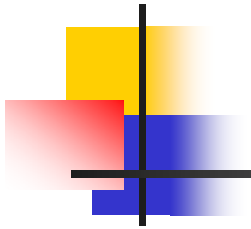


ObCS Element	Feature	Rendering method
Place Planes	token entered <p>	p.show()

4)Rendering



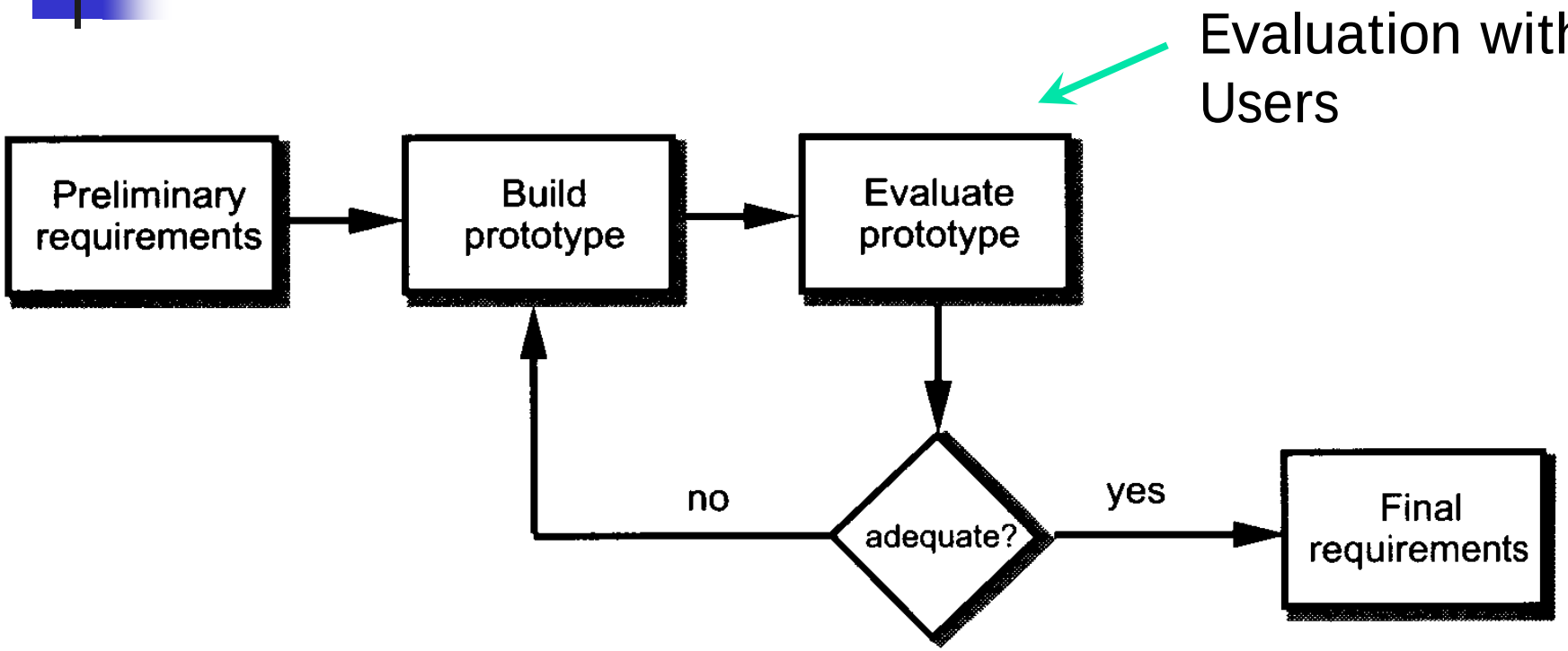
A Short Demo



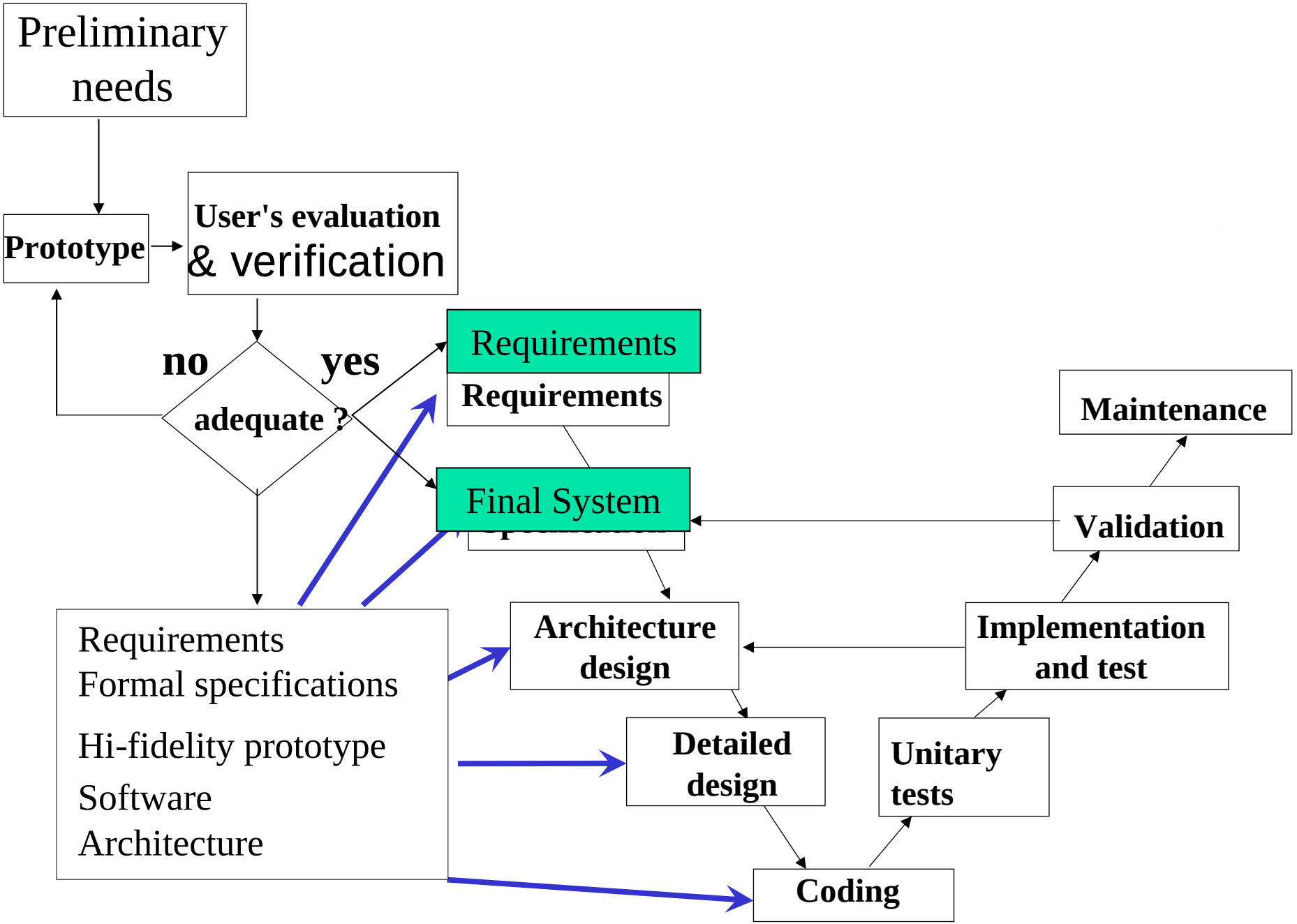
Interactive Systems Models

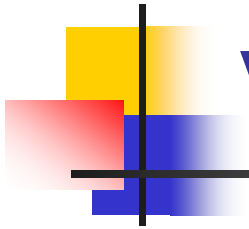
- Task model
  - Abstract description of user's planned activity
- Scenario
  - Concrete description of user's observed activity
  - Sequence of actions performed to achieve a goal
- System model
  - Structural (object-oriented) and behavioral description
  - How user actions change the state of the system
  - How the state of the system:
    - limits legal user actions
    - rendered to the user

Throw-Away Prototyping



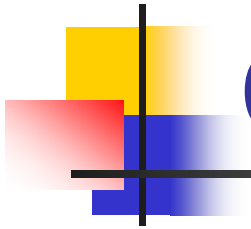
William Newman (1999)


Preliminary needs
Prototype
User's evaluation & verification
no
yes
adequate ?
Requirements
Requirements
Final System
Maintenance
Validation
Requirements
Formal specifications
Hi-fidelity prototype
Software Architecture
Architecture design
Implementation and test
Detailed design
Unitary tests
Coding



Verification

- What ?
  - Safety “nothing bad will ever happen”
  - Liveness “something good will eventually happen”
  - “good cooperation between Models”
  - Efficiency (both local and global)
- How ?
  - Based on Petri nets theory (the basic brick of our system modelling technique)
  - Cross execution of models
  - Performance prediction techniques



Conclusion

- A first step towards a method for the construction of safety critical Interactive Systems (usability and safety)
- Tool-support as a central concern for the applicability of the method
- Mature notations and tools
- Apply the notation and tools to other domains
  - Military systems, satellite control
  - Business applications