

Checking UML Dynamic Diagrams: Synchronous Interface Behavior

C. André

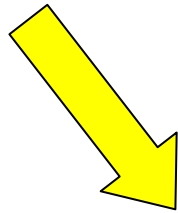
M-A. Peraldi-Frati

J-P. Rigault

Introduction

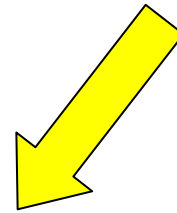
Sequence
Diagrams

simplicity
legibility

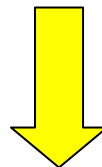


Message Sequence
Charts

structure
semantics



**Synchronous Interface
Behaviors**



Esterel programs

SIB: Features

- Syntax

- Graphical language
- Block-structured
- Restricted construct set

- Semantics

- Mathematical semantics
Synchronous process algebra

a SIB $\xrightarrow{\text{Semantically equivalent}}$ an Esterel program

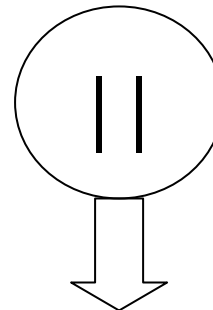
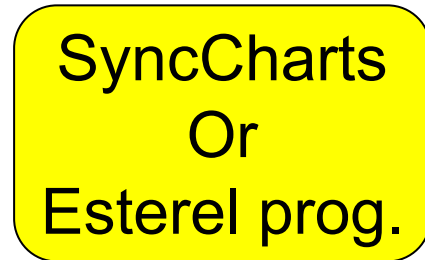
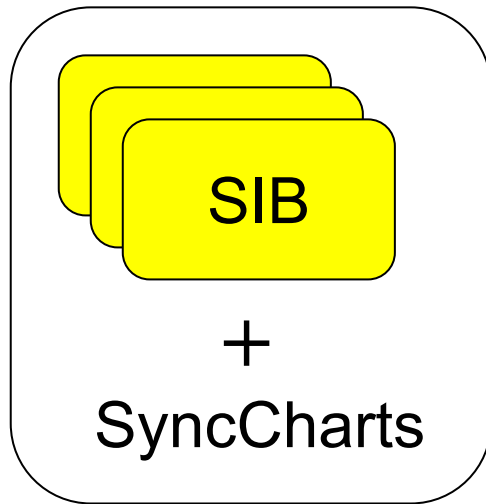
SIB: What for?

- Expressing expected behaviors
 - Discrete-event model
 - Partial ordering of events (signals)
 - Temporal constraints
- Interactive simulation
- Property checking

SIB: How?

Expected behavior

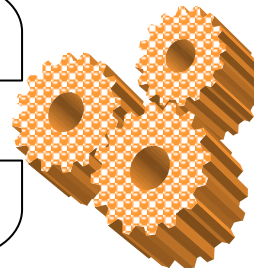
Controller (to check)



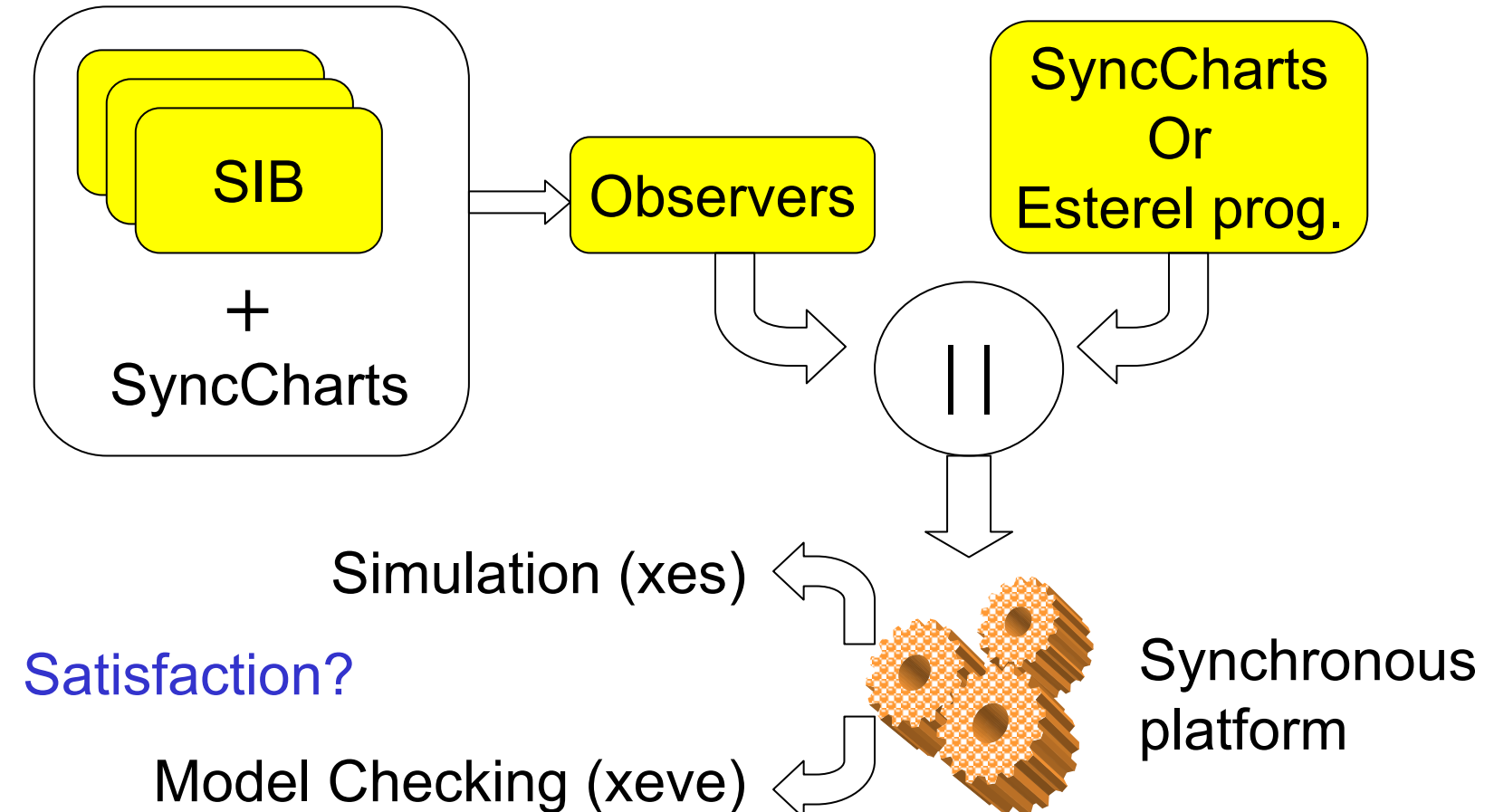
Simulation (xes)

Satisfaction?

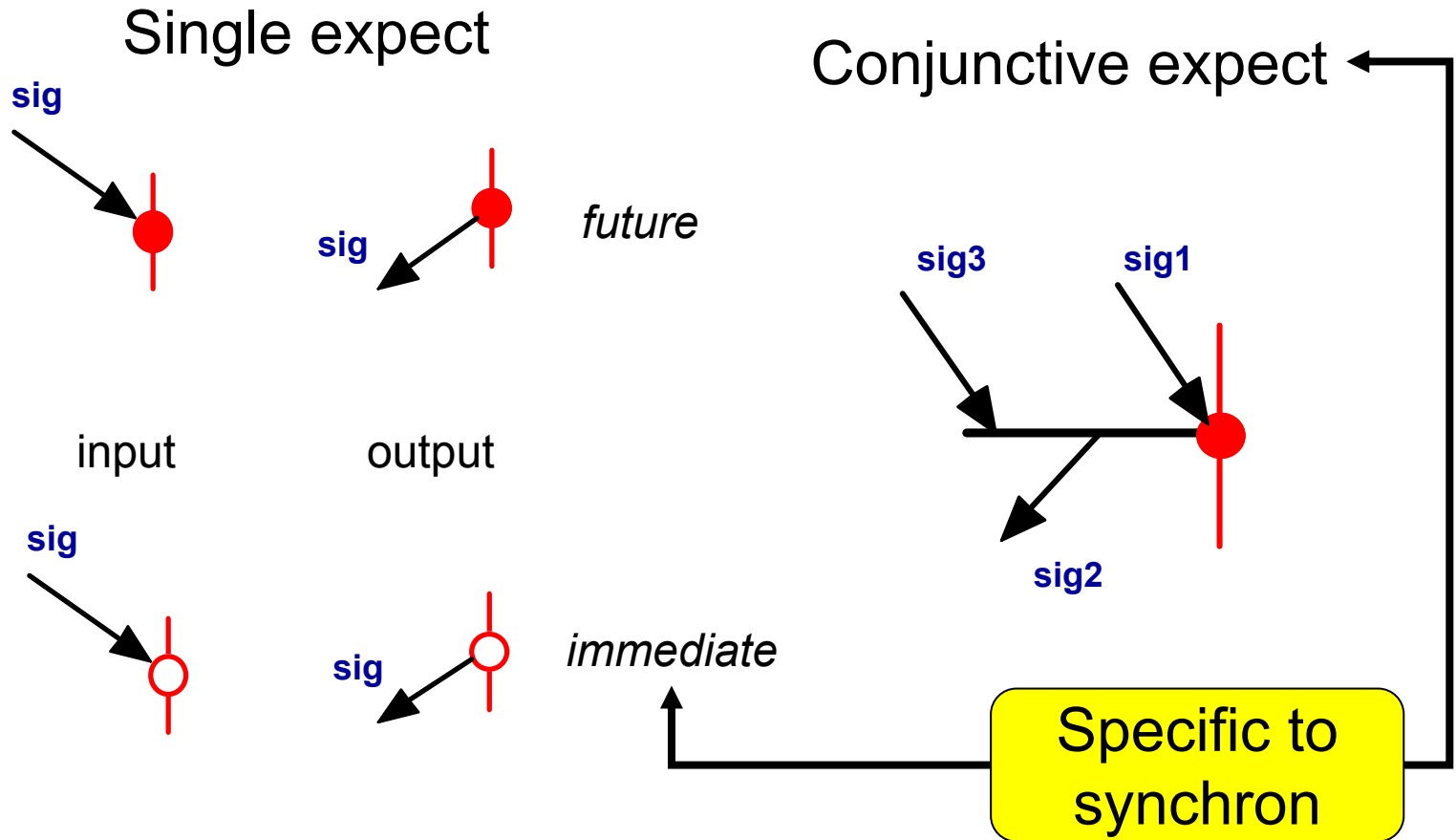
Model Checking (xeve)



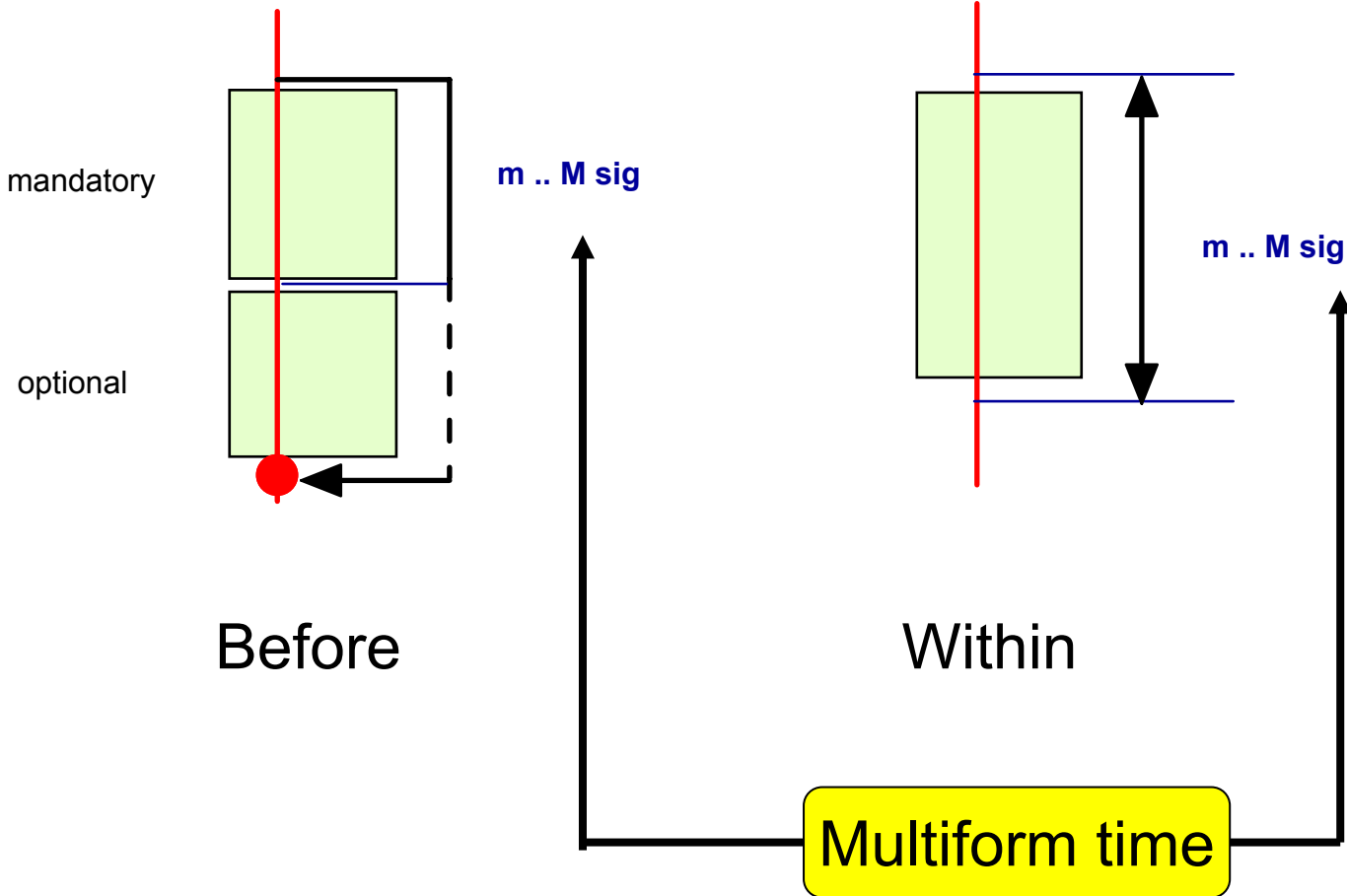
Synchronous
platform



Syntax: Expect

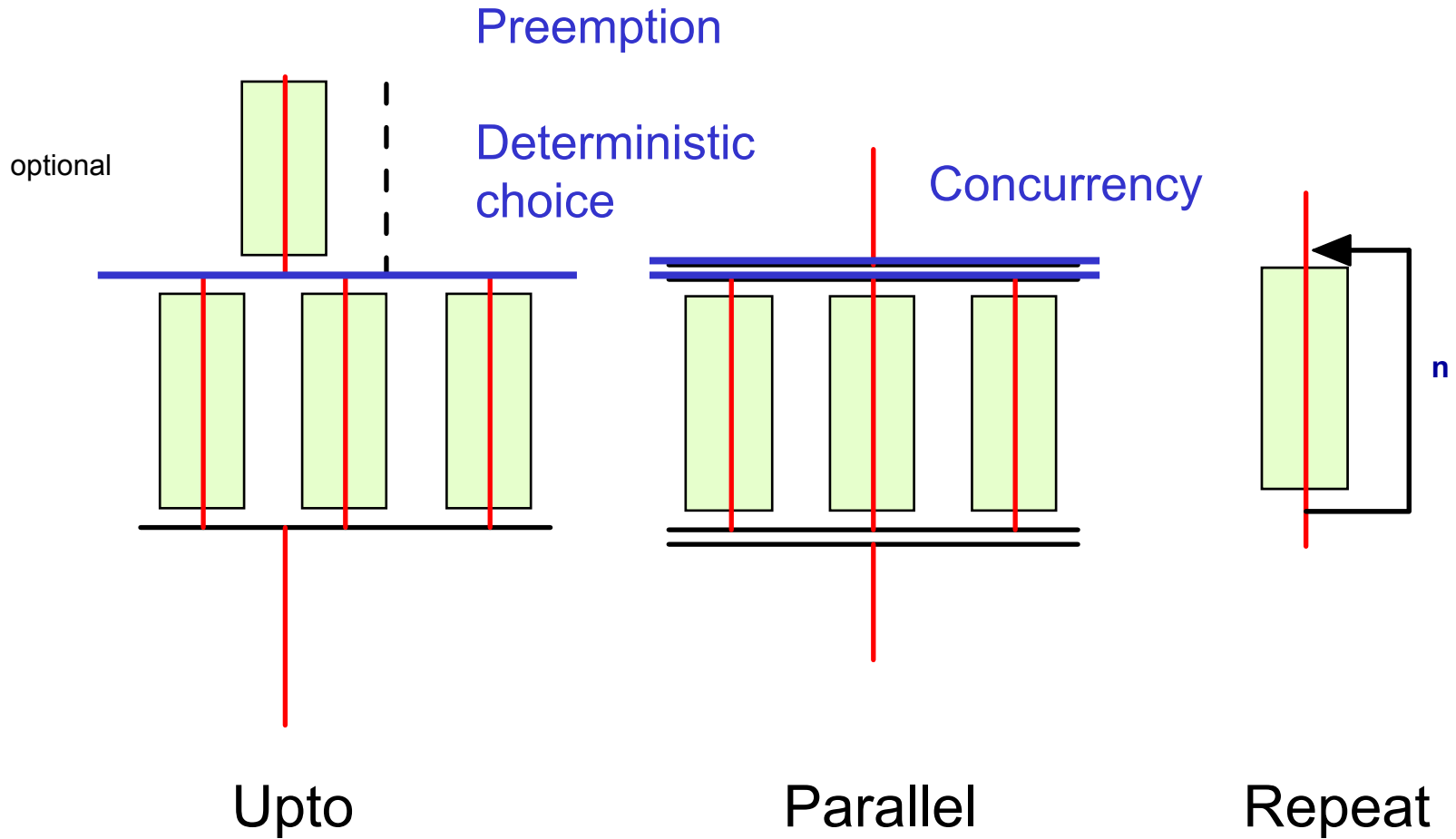


Syntax: Temporal Constraints



Specific system events chosen as time-bases

Syntax: Other Constructs



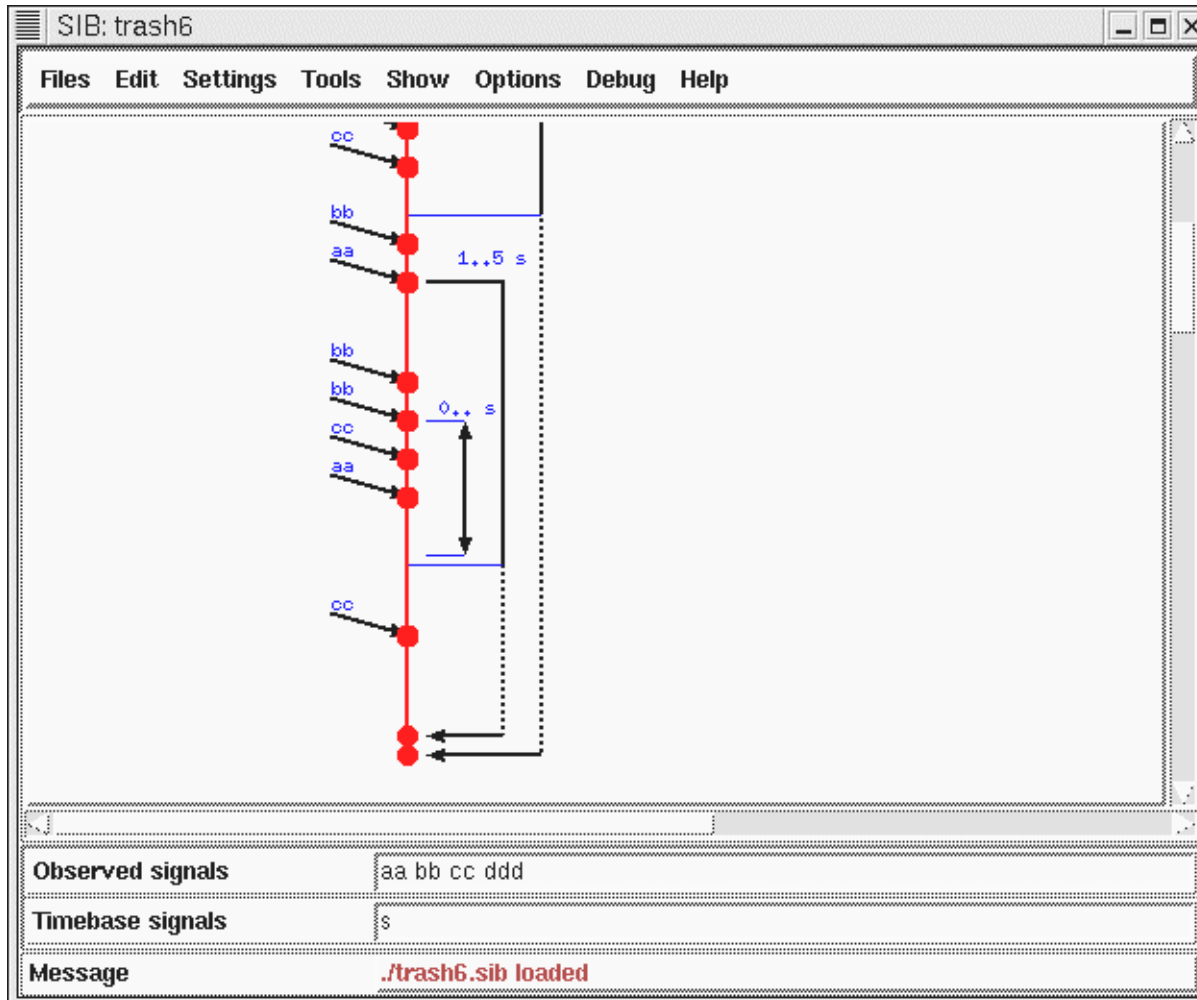
Implementation

- Prototype under construction (Tcl-Tk)
 - Graphical editor
 - Compiler to Esterel code
 - Automated code instrumentation for
 - Interactive simulation
 - Model checking
 - Circuit generation (blif)
- Takes full advantage of the Esterel S/W environment

Conclusion

- User-friendliness (graphic interface)
- Powerful constructs to express scenarios + properties
(no margin annotations!)
- Precise semantics, in terms of a synchronous process calculus
- Integration within the Esterel synchronous platform

Screen Snapshots



Screen Snapshots

