Universidade do Minho
Escola de Engenharia

Logic controllers dependability verification using a plant model

José M. MACHADO [(1)], Bruno DENIS [(2)], Jean-Jacques LESAGE [(2),] Jean-Marc FAURE [(2)], Jaime C. L. FERREIRA DA SILVA [(1)]

[(1)] Mechanical Engineering Department, University of Minho, PORTUGAL

[(2)] LURPA, Ecole Normale Supérieure de Cachan, FRANCE

DESDES 2006
Rydzyna, Poland; September 26-28, 2006

3rd IFAC Workshop on Dicrete-Event System DESign

Outline

Context and objective of the work

DES modeling for verification purposes

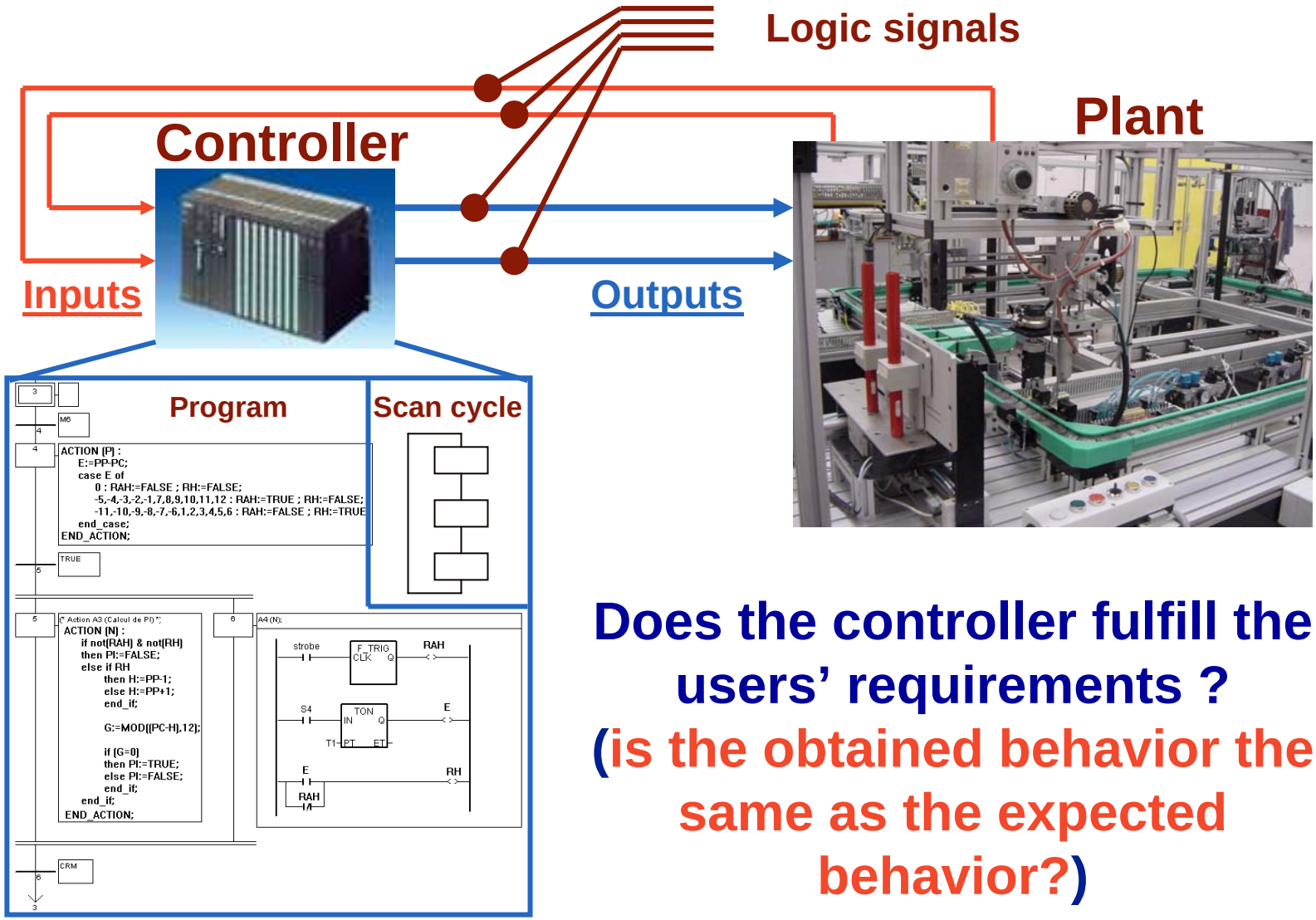
- Modeling complex plants
- Modeling closed loop DES

Case study

Formal verification results

Conclusions and prospects

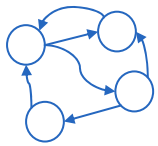
Automated systems considered



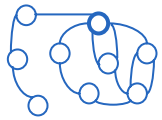
Does the controller fulfill the users' requirements ?
(is the obtained behavior the same as the expected behavior?)

Formal verification using model-checking

Formal representation of a system (DES model)



Satisfies ?



AG ((dp_head_motor_up) =>
EF (!dp_head_motor_up))

Formal representation of a property (temporal logic, observer automaton)

Model – checker
(exhaustive exploration of the state space of the DES model using a symbolic representation)

Property proved or counterexample

in case of negative proof

The model to be verified is obtained either from the controller only (Non Model-Based approach) or from the whole system (Model-Based approach)

Non Model-based and Model-based Model-Checking

- Non (Plant) Model Based (NMB) [Probst *et al.* 1997] [Bornot *et al*. 2000] [Mertke & Frey, 2001] [Rossi, 2004]

Focus is put on properties proof of the **isolated controller**

- (Plant) Model-Based (MB) [Kowalewski & Preuβig,1996] [Raush & Krogh, 1998] [Zaytoon & Carré-Ménétrier, 1999] [Mertke & Menzel, 2000] [Hanisch, 2006]

Focus is put on properties proof of the **closed loop system**

- Constraints-based approach: simplified version of MB

Non Model-Based and Model-Based Model-Checking

Which approach must be chosen ?

- NMB:
  - Properties are proved whatever the values of the controller inputs
  - All the plant behaviors (realistic and unrealistic) are considered
- MB:
  - Properties are proved only for realistic combinations of the controller inputs
  - An other model (plant model) is mandatory
  - Building a plant model is a difficult task for large scale industrial systems

These approaches have never been compared

DES modeling for verification purposes

Proposition of a **method** to build complex plant models

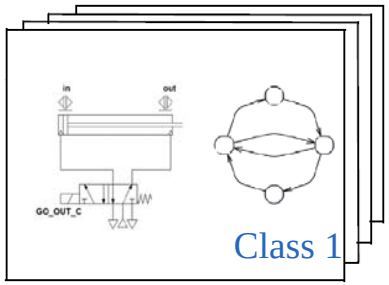
- Generic and modular
- As automated as possible
- Based on a suitable class of automata

Proposition of **algorithms** to obtain the “ready-to-check” model of the closed-loop system

- Aim: **to avoid constructing explicitly the model of the whole system by using automata products** that always lead to combinatory explosion and often include unexpected behaviors
- Main feature: set of **communicating sequencers** (plant, controller and system sequencers)

Building complex plant models

Generic models



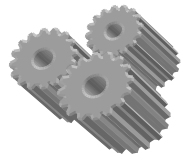
Start

Declaration

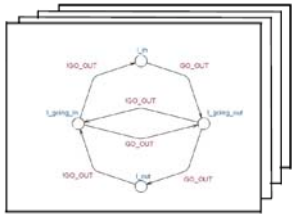
List of the plant modules

Cylinder1 := class1(..., ...)
... := class...(..., ...)
Sensor1 := class...(..., ...)
... := class...(..., ...)
... := class...(..., ...)

Instantiation

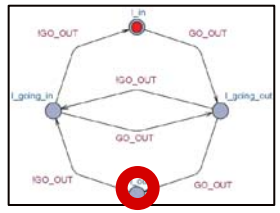


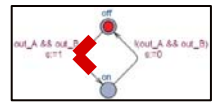
Instantiated models



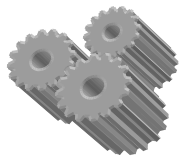
Coordination and specific models (if necessary)

Coordinated models: basic version

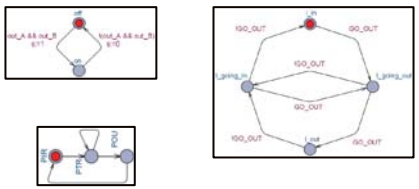




Generation of the complete plant model:
-Plant sequencer
-Models enrichment



Complete plant model



End

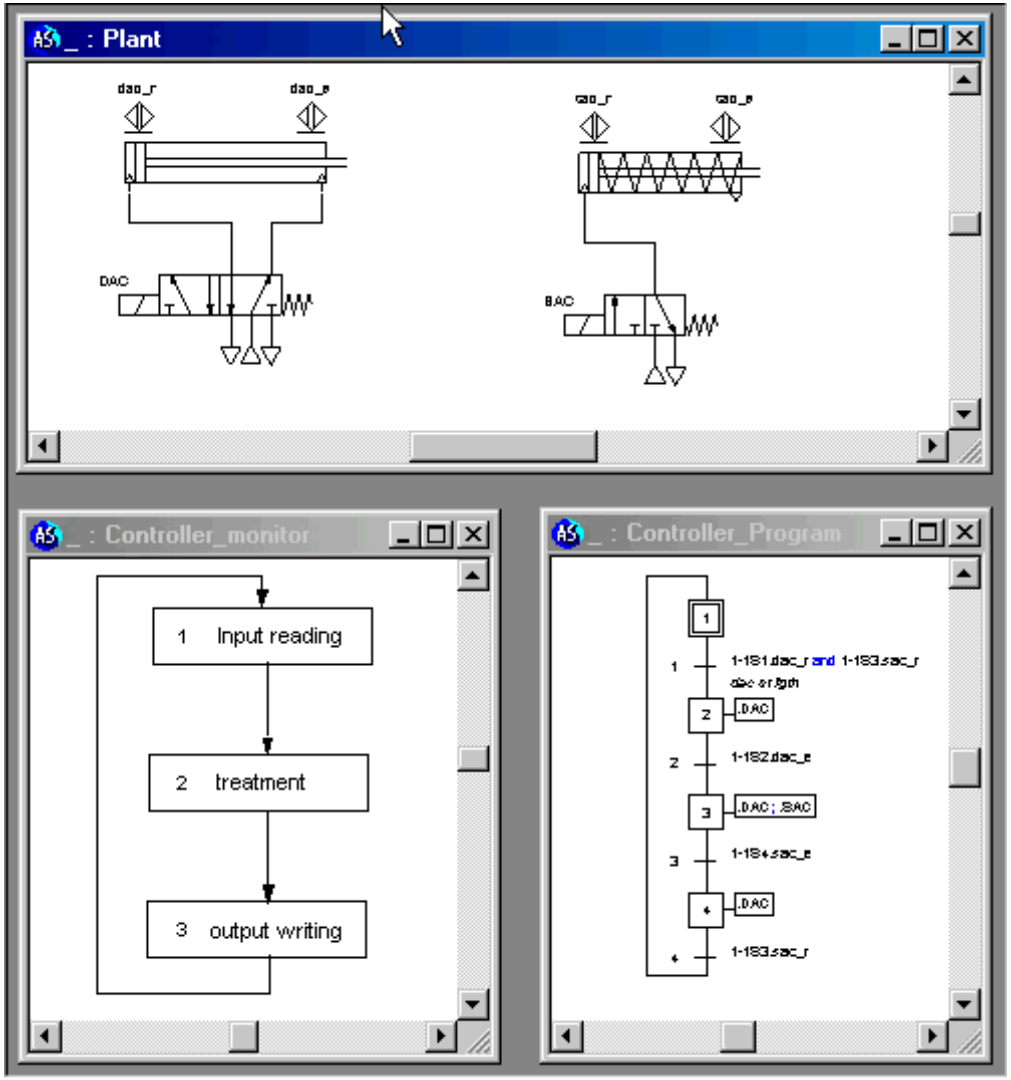
Modeling formalism

Non deterministic and communicant automata (inspired by [Bengston and Yi, 2004])

- An automaton *A* is a triplet <N, n_0, $\sum$> where :
  - N is a finite set of states;
  - $n_0 \in$ N is the initial state;
  - E $\subseteq$ N x τ x $\sum$ x N, is the set of transitions, with
  - τ being a set Boolean expressions defined on the set of logic variables *V*, and
  - $\sum$ a partition of the set of assignments on *V*;

- An automata network {A_1, … A_n} is a set of automata such as:
  - Boolean expressions τ associated with the transitions are defined on the same set of logic variables *V*;
  - Within the network the evolutions are asynchronous;
  - At each instant only one transition of one automaton can be fired;
  - Communication between automata is based on variable sharing.

Taking into account physical behaviors



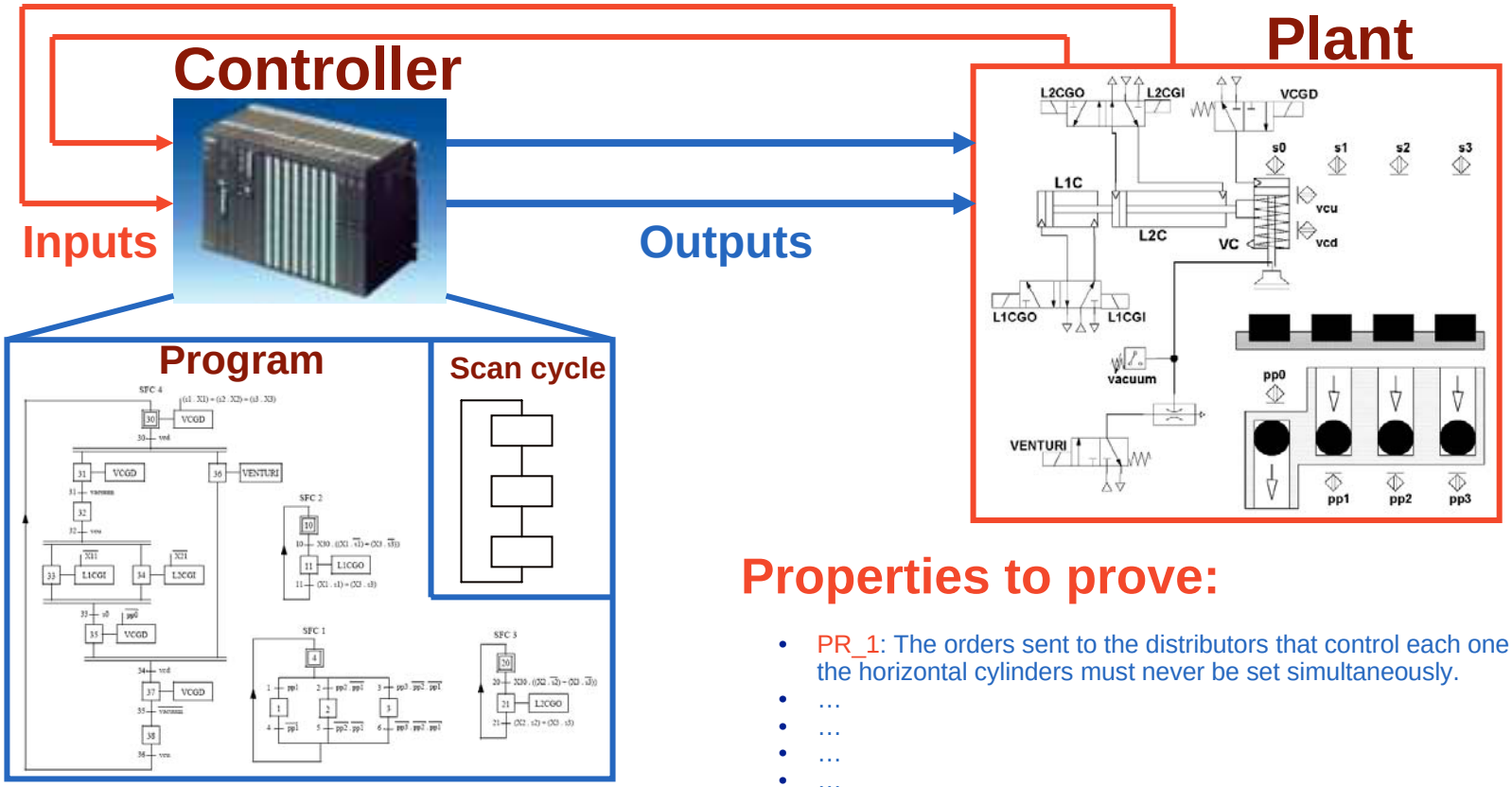
Controller evolution faster than plant evolution

Asynchronous evolutions between controller and plant;

and

Synchronization between plant and controller during the inputs reading and outputs updating phases

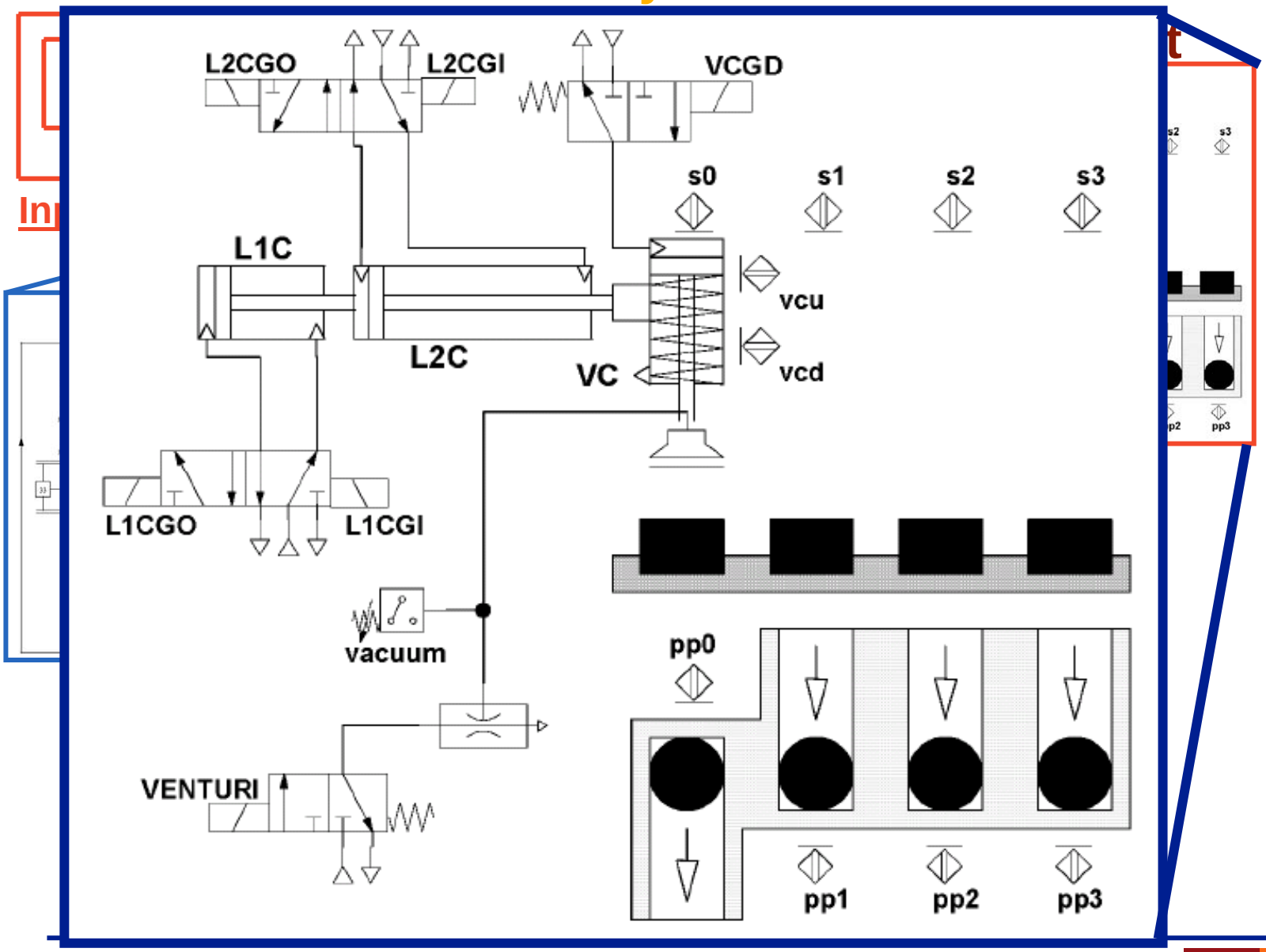
System overview



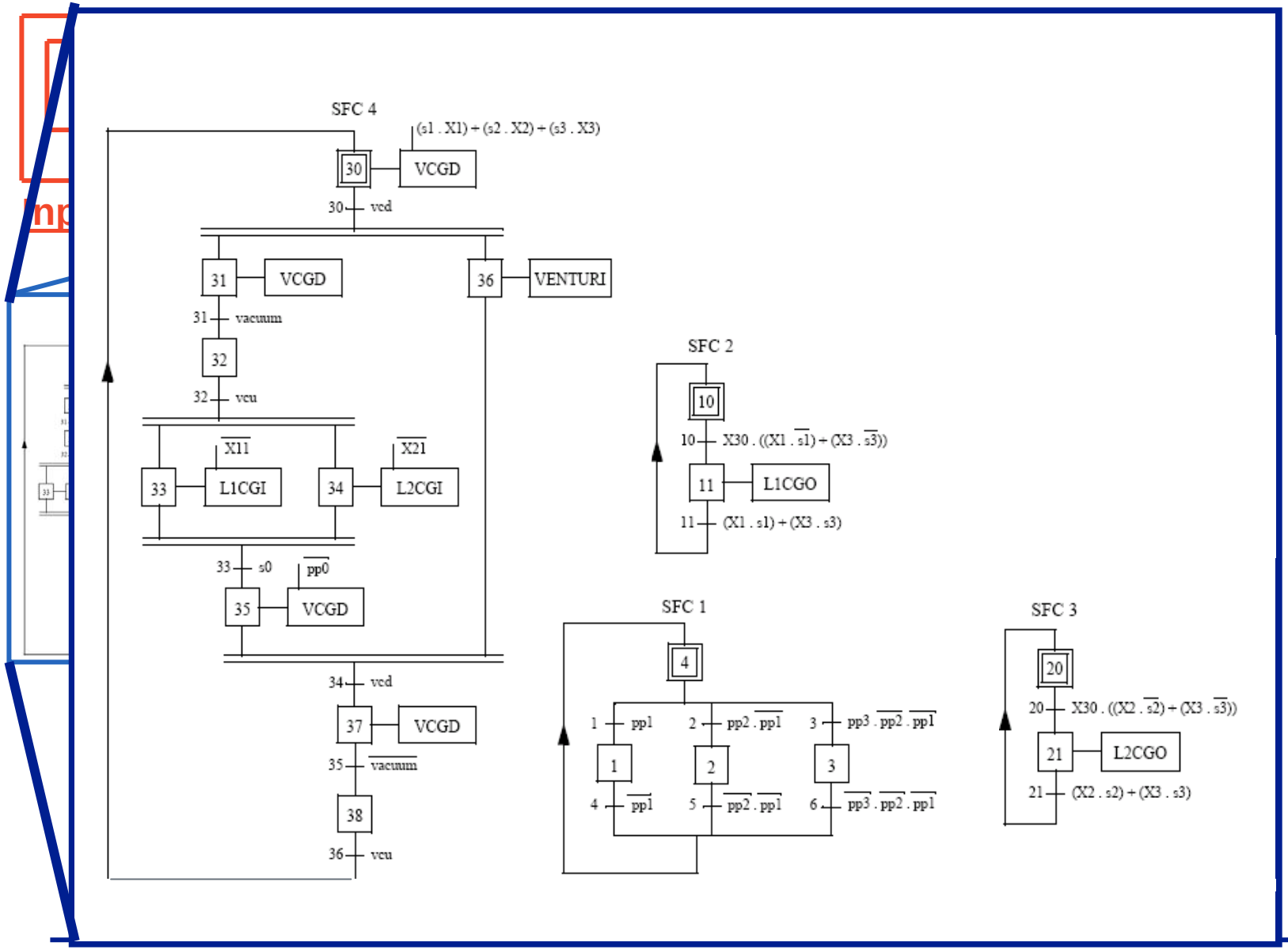
Properties to prove:

- PR_1: The orders sent to the distributors that control each one of the horizontal cylinders must never be set simultaneously.
- …
- …
- …
- …
- …
- PR_7: If a part is detected by a sensor ppi (i∈[1,2,3], then, in the future, the rod of one of those cylinders will move outwards
- …
- …
- …

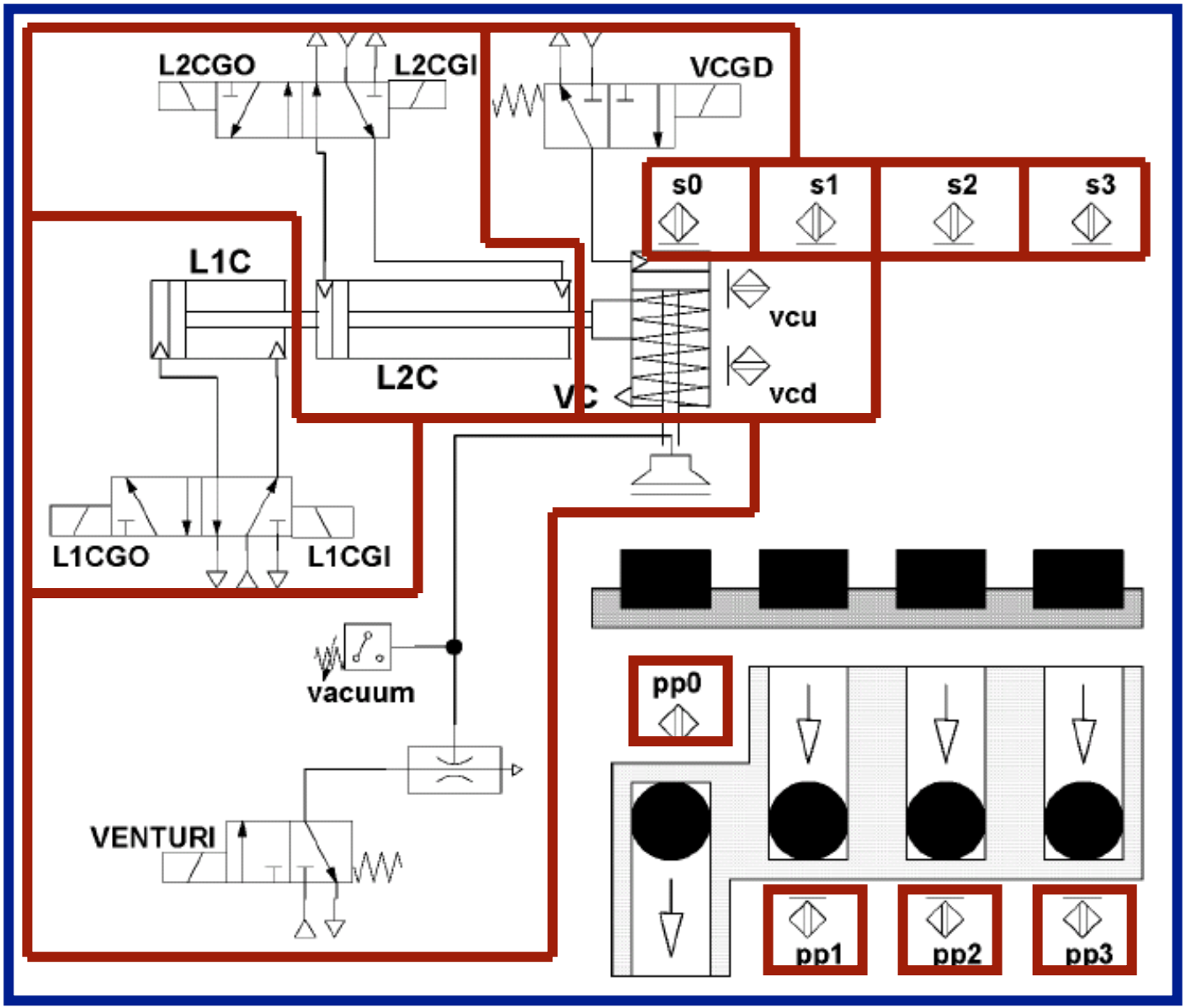
Plant layout



Controller program



Modeling the plant: considered modules



Plant model construction

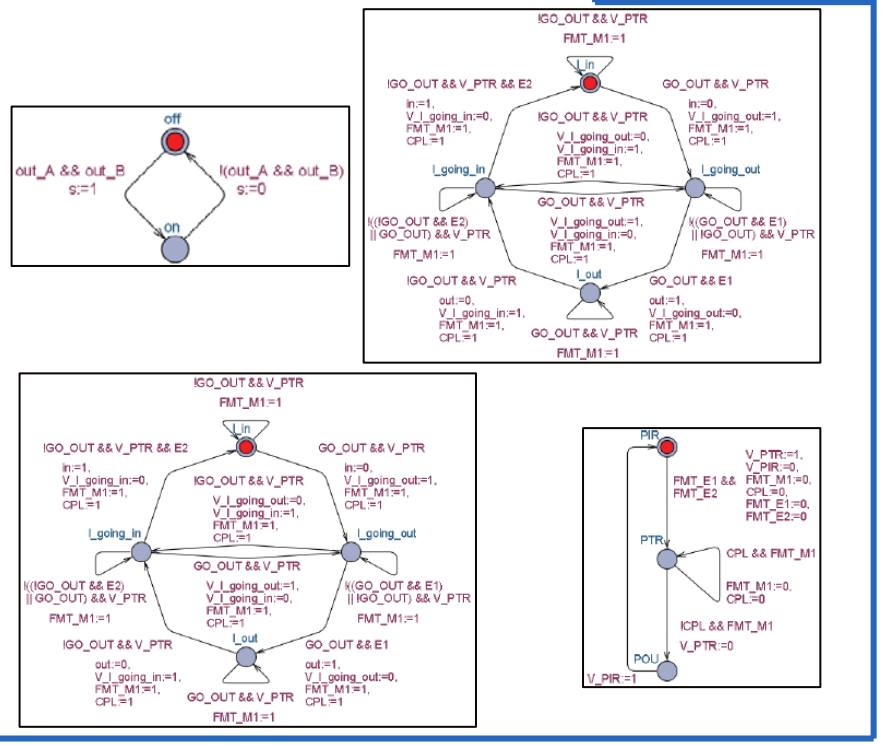
According to the 4 steps method presented above

Result: a set of communicating automata and one plant model sequencer (automatically constructed)

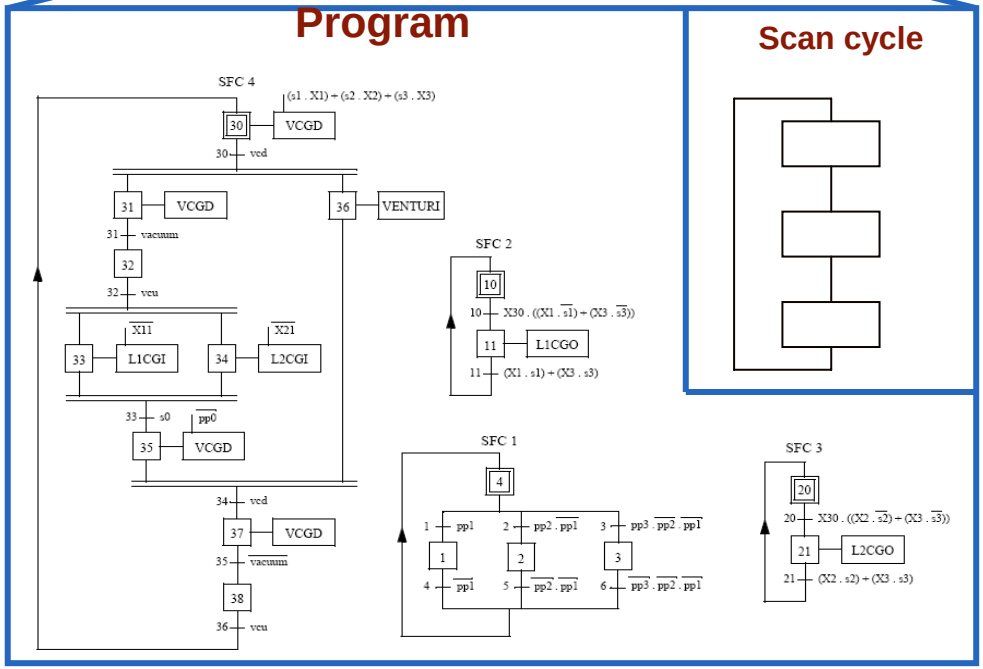
Interest of this approach:

no automata product is performed, only sequencing of automata is necessary

Extract of the plant model

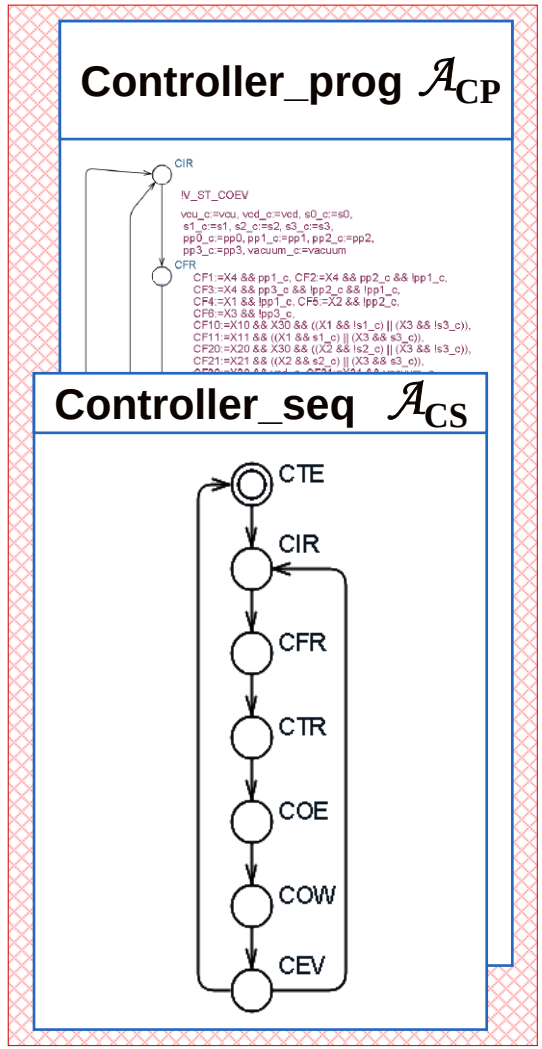


Controller model

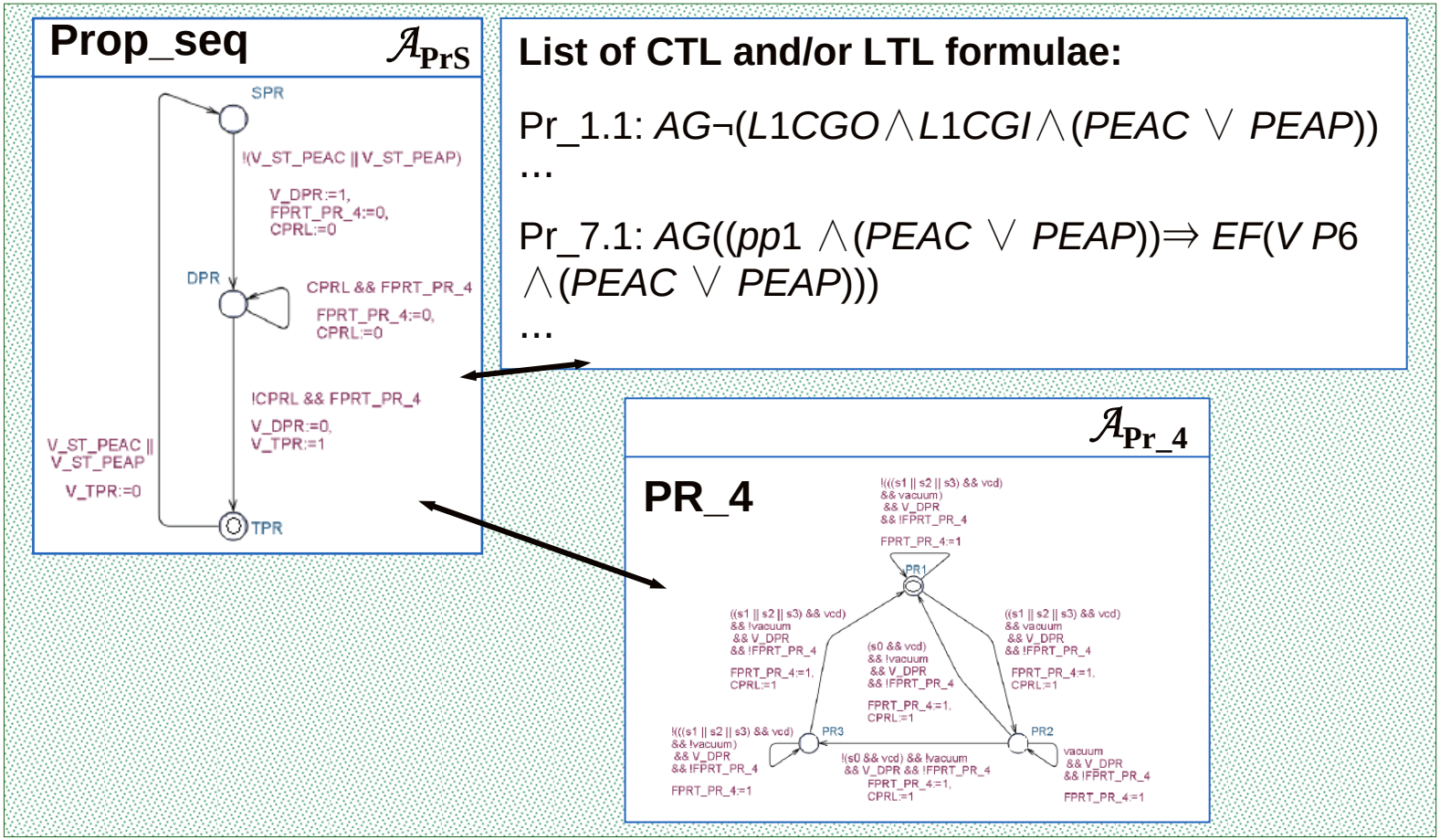


Automatic Translation

Via algebraic modeling



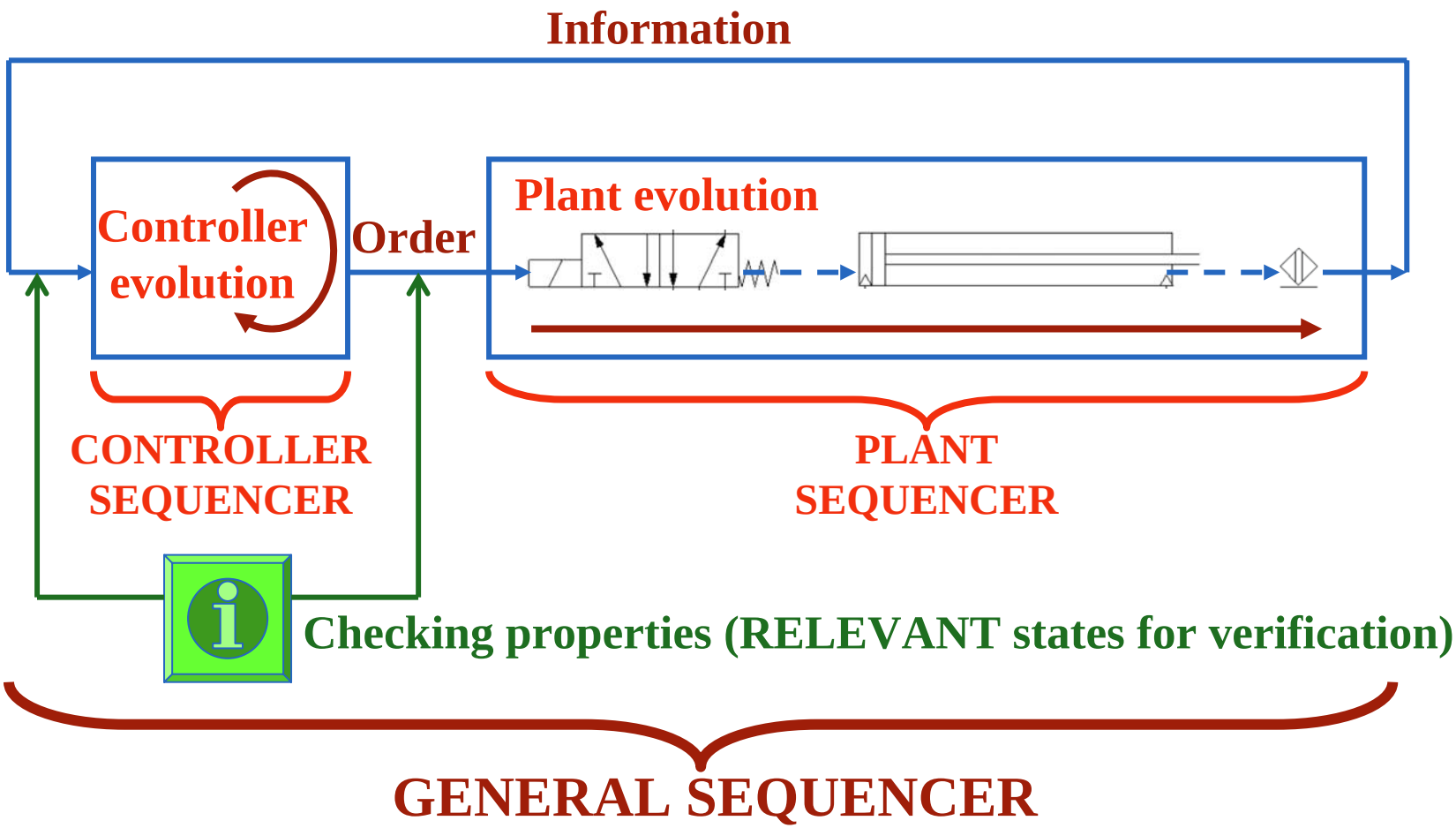
Properties models



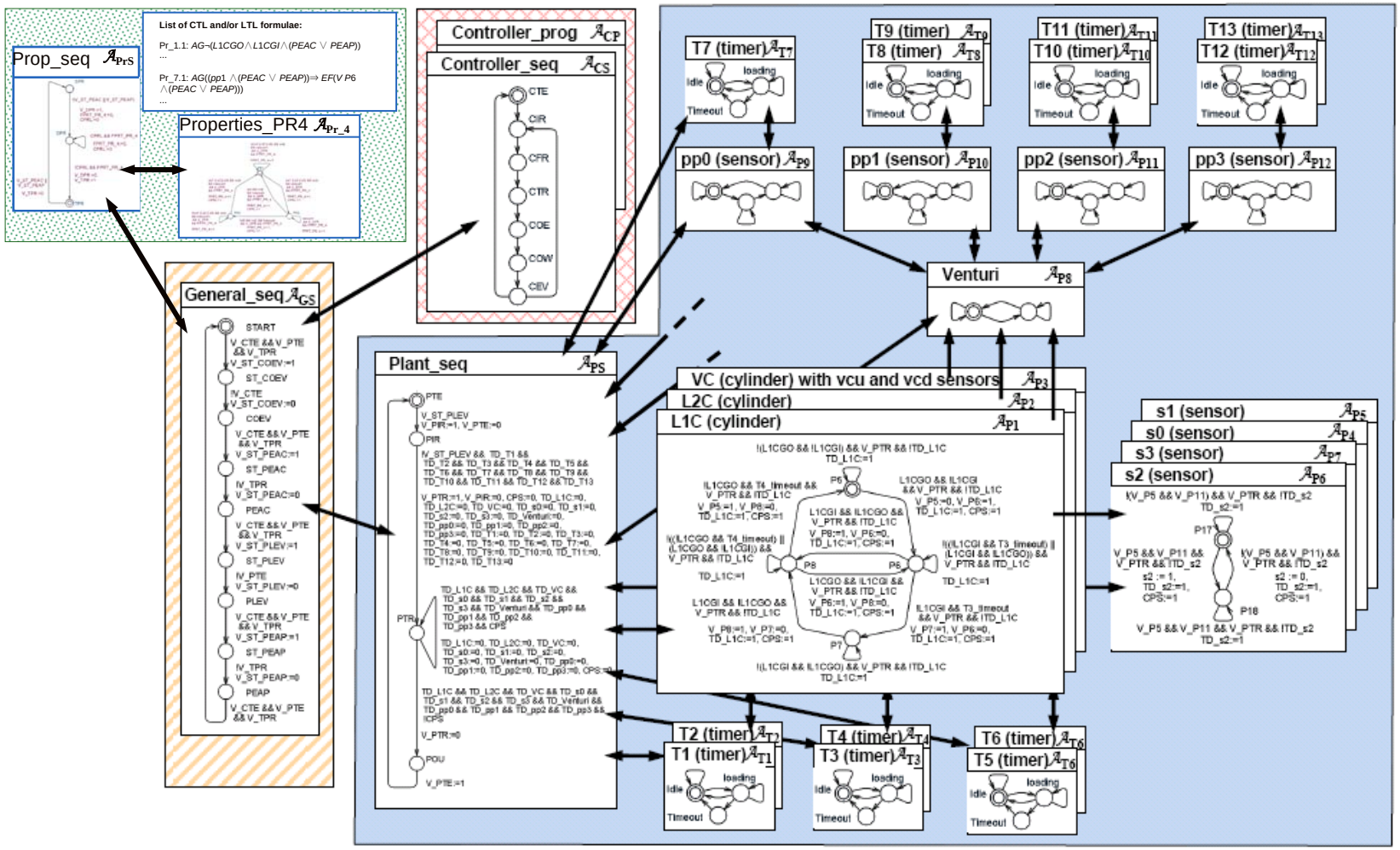
Two kinds of properties: Safety and Liveness properties

- the two opposite commands of the horizontal cylinder must never be simultaneously set
- the manipulator takes the part at the picking position and doesn't release it until the placing position is reached

Modeling the closed-loop system

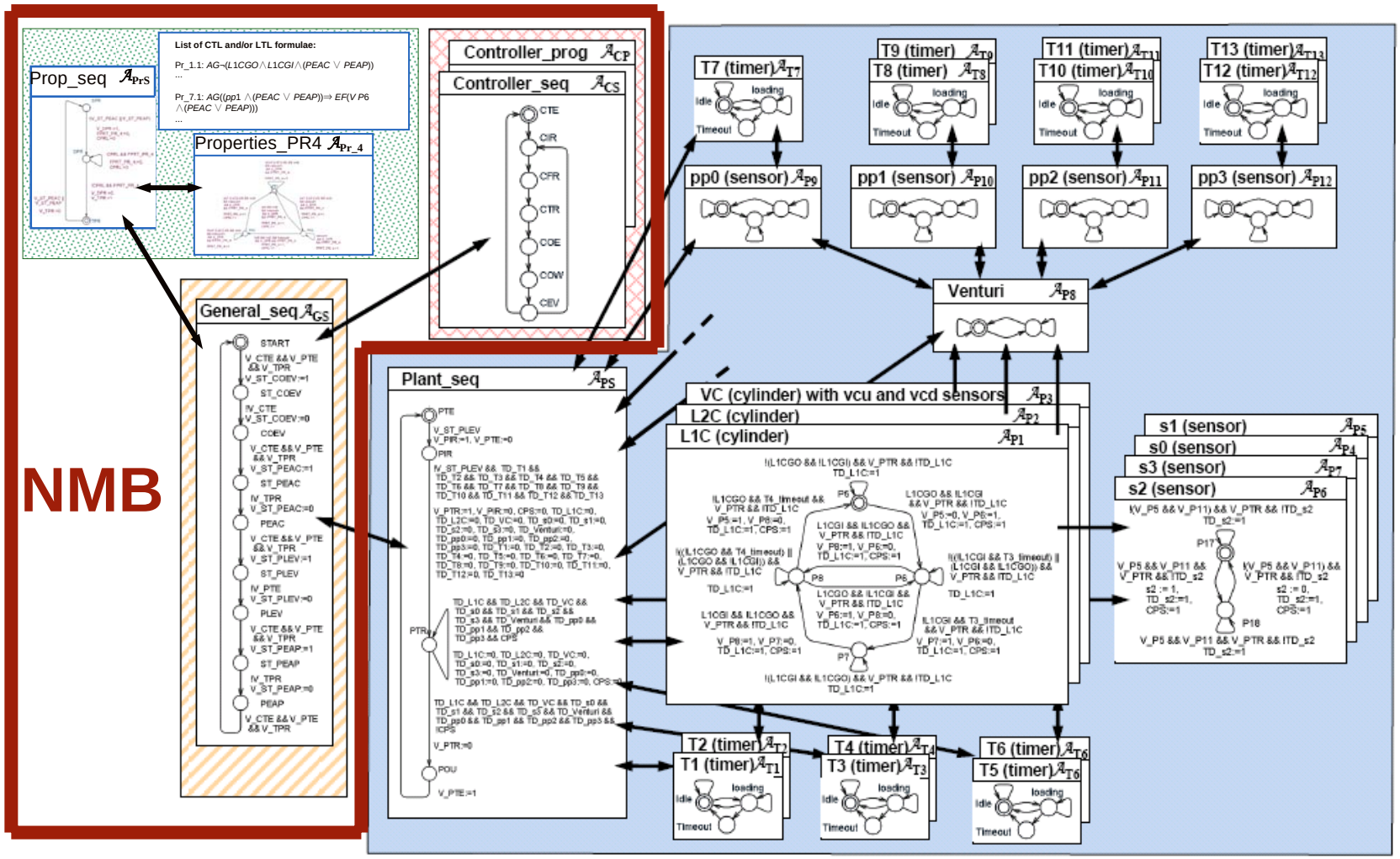


Overall model

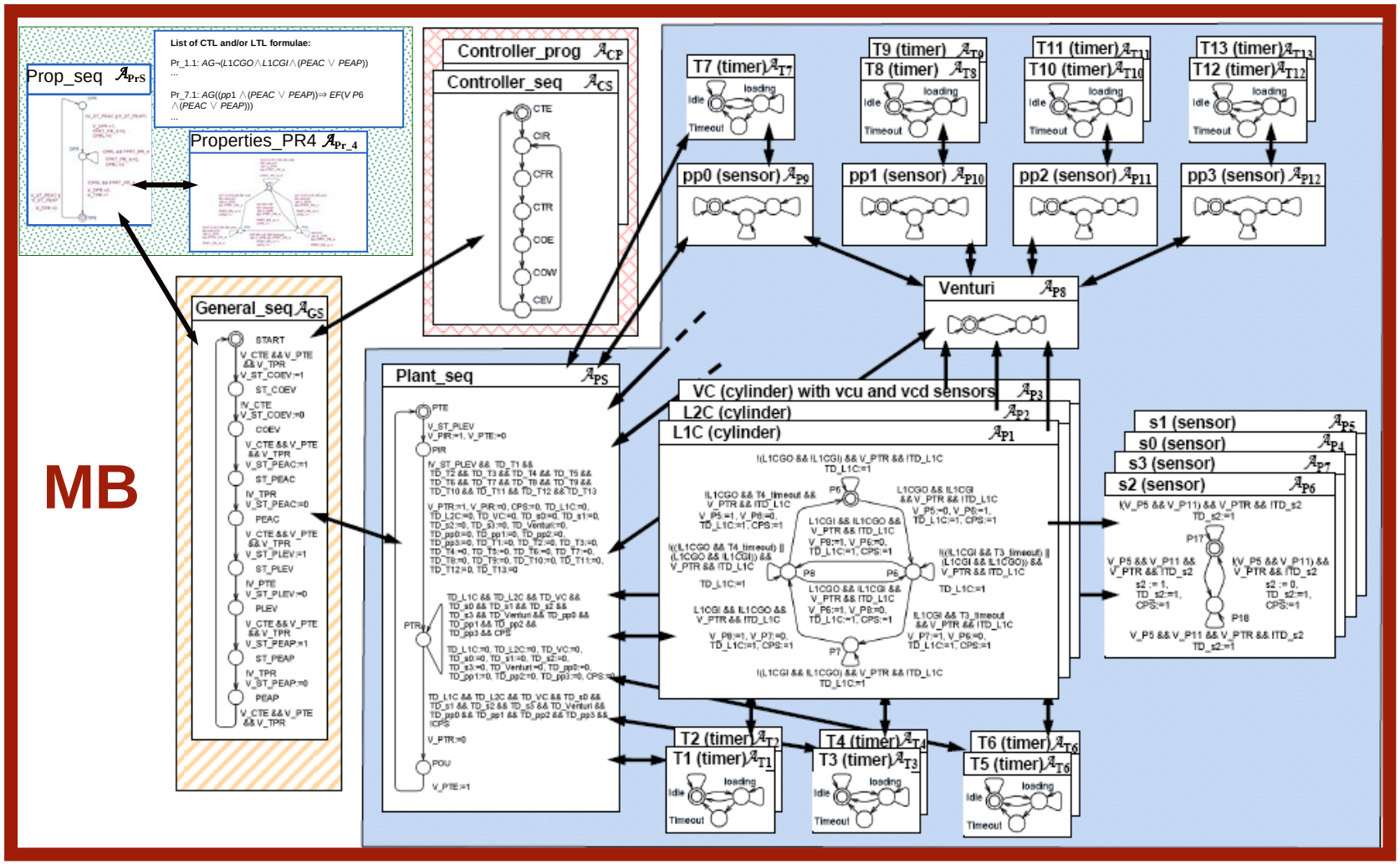


42 safety and liveness properties have been verified with NuSMV running on a computer with a P4 CPU at 2.54GHz and 2GB RAM

Using the overall model for formal verification



Using the overall model for formal verification



Case of identical answers

Prop	Type	NMB	MB
PR_1	Safety	true	true
PR_2	Safety	true	true
PR_3	Safety	false	true
PR_4	Safety	false	true
PR_5	Safety	X	true
PR_6	Liveness	true	true
PR_7.1	Liveness	X	true
PR_7.2	Liveness	X	true
PR_7.3	Liveness	X	true
PR_8.1	Liveness	true	true
PR_8.2	Liveness	true	true
PR_8.3	Liveness	true	true
PR_9	Safety	X	true

The approach that yields the more meaningful results must be chosen

X means ‘absence of answer’

Case of contradictory answers for safety properties

Prop	Type	NMB	MB
PR_1	Safety	true	true
PR_2	Safety	true	true
PR_3	Safety	false	true
PR_4	Safety	false	true
PR_5	Safety	X	true
PR_6	Liveness	true	true
PR_7.1	Liveness	X	true
PR_7.2	Liveness	X	true
PR_7.3	Liveness	X	true
PR_8.1	Liveness	true	true
PR_8.2	Liveness	true	true
PR_8.3	Liveness	true	true
PR_9	Safety	X	true

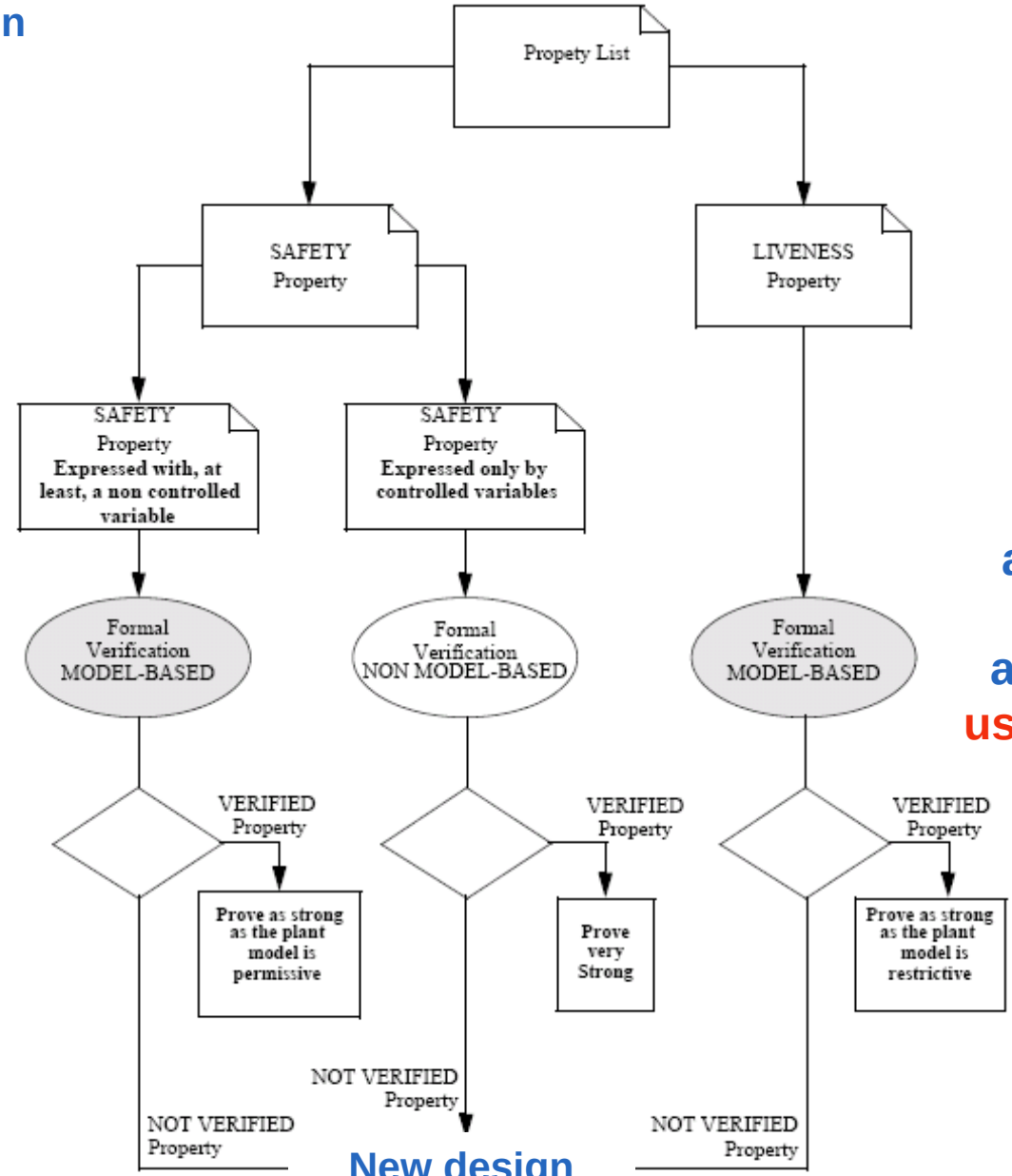
Requires complementary analysis

Absence of answer with the NMB approach

Prop	Type	NMB	MB
PR_1	Safety	true	true
PR_2	Safety	true	true
PR_3	Safety	false	true
PR_4	Safety	false	true
PR_5	Safety	X	true
PR_6	Liveness	true	true
PR_7.1	Liveness	X	true
PR_7.2	Liveness	X	true
PR_7.3	Liveness	X	true
PR_8.1	Liveness	true	true
PR_8.2	Liveness	true	true
PR_8.3	Liveness	true	true
PR_9	Safety	X	true

The property can be proved only by taking into account the plant behavior

Formal verification results



New design of the controller

PROPOSITION:

Decision tree taking into account the kind of properties and the variables used in the property

Conclusions and prospects

Proposition of a method to build complex plant models

Proposition of algorithms to obtain the "ready-to-check" model of the closed-loop system without combinatory explosion nor unexpected behaviors

Proposition of a decision tree to choose the most appropriate approach (NMB/MB)

To apply these techniques to timed systems (using the UPPAAL model-checker)

To introduce faulty behaviors in the plant model so as to check controller behavior in the presence of plant faults

1st IFAC Workshop on

Dependable Control of Discrete Systems

DCDS'07

June 13-15, 2007
Paris - Cachan, France

The aim of this workshop is to provide the communities of safety/reliability analysis and of DES with an opportunity to exchange and to discuss new developments in the field of **dependable control** of discrete event systems