Synthèse et filtrage robuste de la commande pour des système manufacturiers sûrs de fonctionnement

Synthesis and robust filtering of control for dependable manufacturing systems

Ph.D. for the doctor title of the University of Reims Champagne-Ardenne

By Pascale MARANGE

November 05 2008

Supervised by Bernard RIERA (Professor)
and François GELLOT (Assistant Professor)

CONSEIL REGIONAL CHAMPAGNE ARDENNE

CReSTIC
Centre de Recherche en STIC

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Plan

1. Introduction
    1. Works Context
    2. Works in the domain
    3. Contribution of thesis works
2. Control validation approach by filter
    1. Definition of filter model
    2. System adaptation
    3. Method to define the safety constraints
    4. Discussion
3. Safety constraints verification
    1. Problematic
    2. Verification approach
    3. System model
    4. Discussion
4. Pedagogical applications
    1. Context
    2. Real system : Productis Machine
    3. Virtual system : Warehouse, ITS PLC
    4. Discussion
5. Conclusion and future works

Introduction
Work context

- Problematic :
  - How to secure Automated Production Systems (APS)?

- Use of APS raises problem of :
  - Security risks and damage
  - Maintenance time and qualify people
  - Important cost

- In the industrial case, several objectives :
  - Economic: reduced cost of manufacture, reduced cost of stock, maximum production
  - Humans: improving working conditions
  - Technical: reduce production cycles, increase product quality, improve systems flexibility, improve systems availability

- Possible malfunctions :
  - External : modification of customer demand, bad raw materials, breakdown…
  - Internal : material breakdown of calculator, plant wear, programming error…

Introduction
Work context

- An error at level of control programming :
  - A bad operator behavior
  - A bad control programming design

- Due to bad system vision, bad understanding of what to do, or it may be intentional

- It requires to :
  - Assure the plant safety
  - Assure that the specification (technical standard) is conformed
  - Adapt the system vision to the control designer

- Application :
  - Industrial case :
    - Systems more and more complex
    - Reduce the maintenance intervention and the start-up time
  - Pedagogical case :
    - Use Information and Communication Technology : Possibility of equipment collaborative use, remote practical work

Introduction
Work context

- Requirements and ideas:
  - To safe the system :
    - Usable in the real world
    - Adapted and time compatible with the controller
    - Independent from control
  - To adapt the system to control designer by:
    - adapting the difficulty parameter (dimension, synchronization, hierarchization) but keeping the system as a whole
    - giving a comprehensible explanation in case of errors

Introduction
Works in the domain

- Definition of validation and verification terms
  - Validation : is the design conform to specification ?
  - Verification : is the validation correctly defined ?

- To analyze the system and to adapt it
  - System definition according to 2 axis : "Whole-Part" et "Means-End" (Lind, 1994)
  - Hierarchical analysis (Belhimeur, 1989)

- To ensure the control dependability
  - Validation/verification approach : simulation, reachability analysis (Kowalewshi et al., 1996), model checking (Gourguff, 2007 ; Barragan, 2007), Theorem Proving (Volker, 2002; Roussel 2002)
  - Monitoring approach (Zamaï, 1997 ; Lhoste, 1994 ; Cruette, 1991)
  - Synthesis approach (Ramadge, 1989 ; Wonham, 1987 ; Tajer, 2006)

Introduction
Works in the domain - in CReSTIC

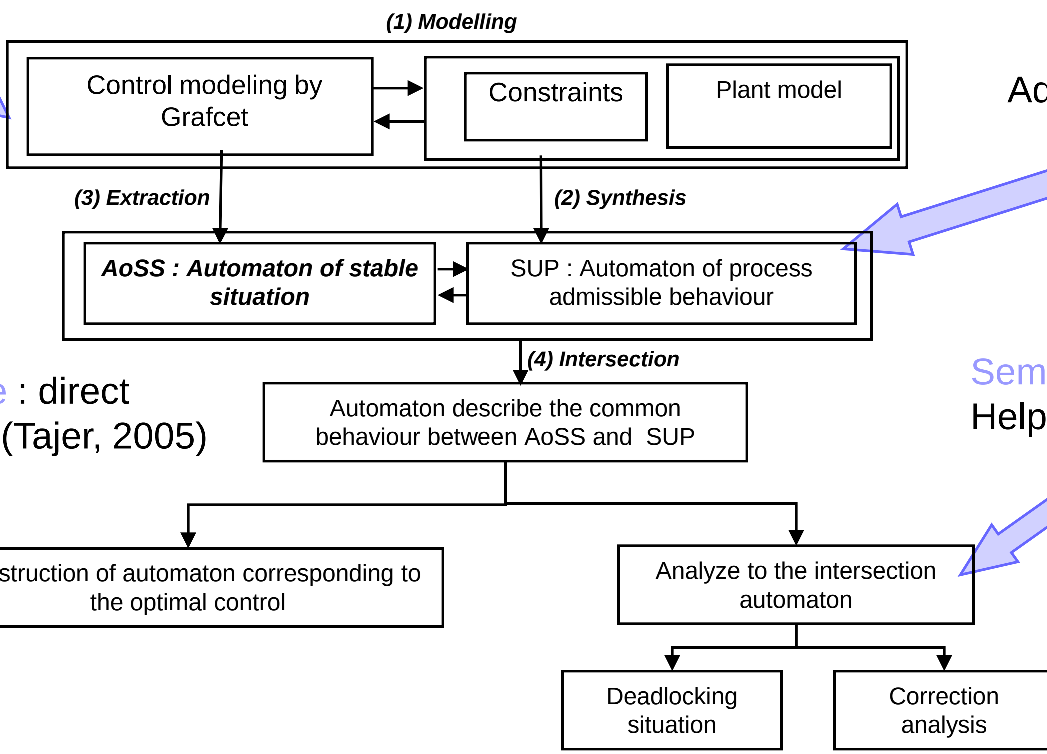
- CReSTIC : Control synthesis works to get a sure determinist and no deadlocking control
- New orientation : support system for the control designer

Plant, control and constraints modeling (Philippot et al., 2004)

Adaptation of Kumar synthesis approach (Kumar, 1991)

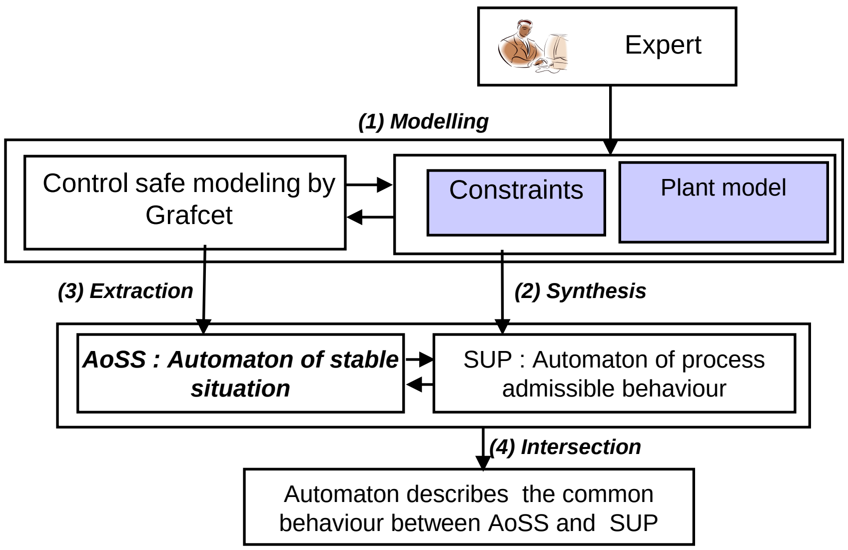
Automatic mode : direct implementation (Tajer, 2005)

Semi-automatic mode : Help to control design

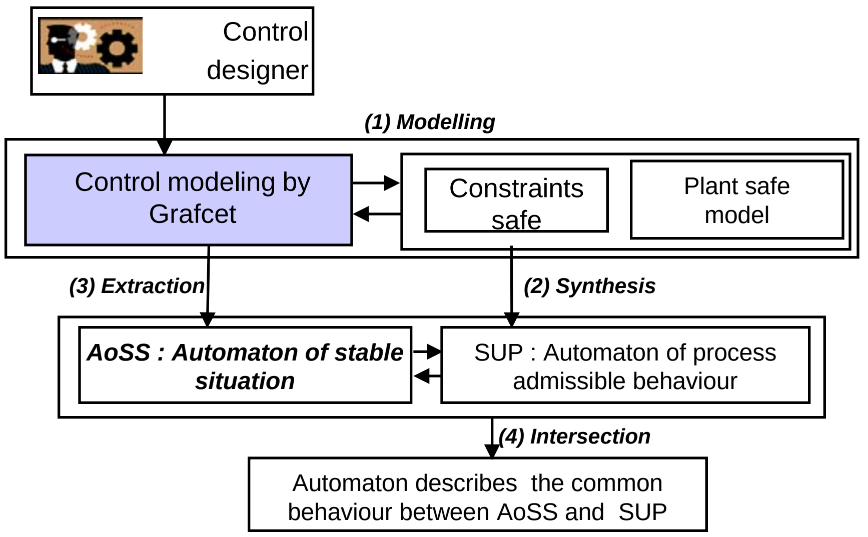


Introduction
Works in the domain - in CReSTIC

- First step : Plant model and constraints verification



- Second step : Control validation



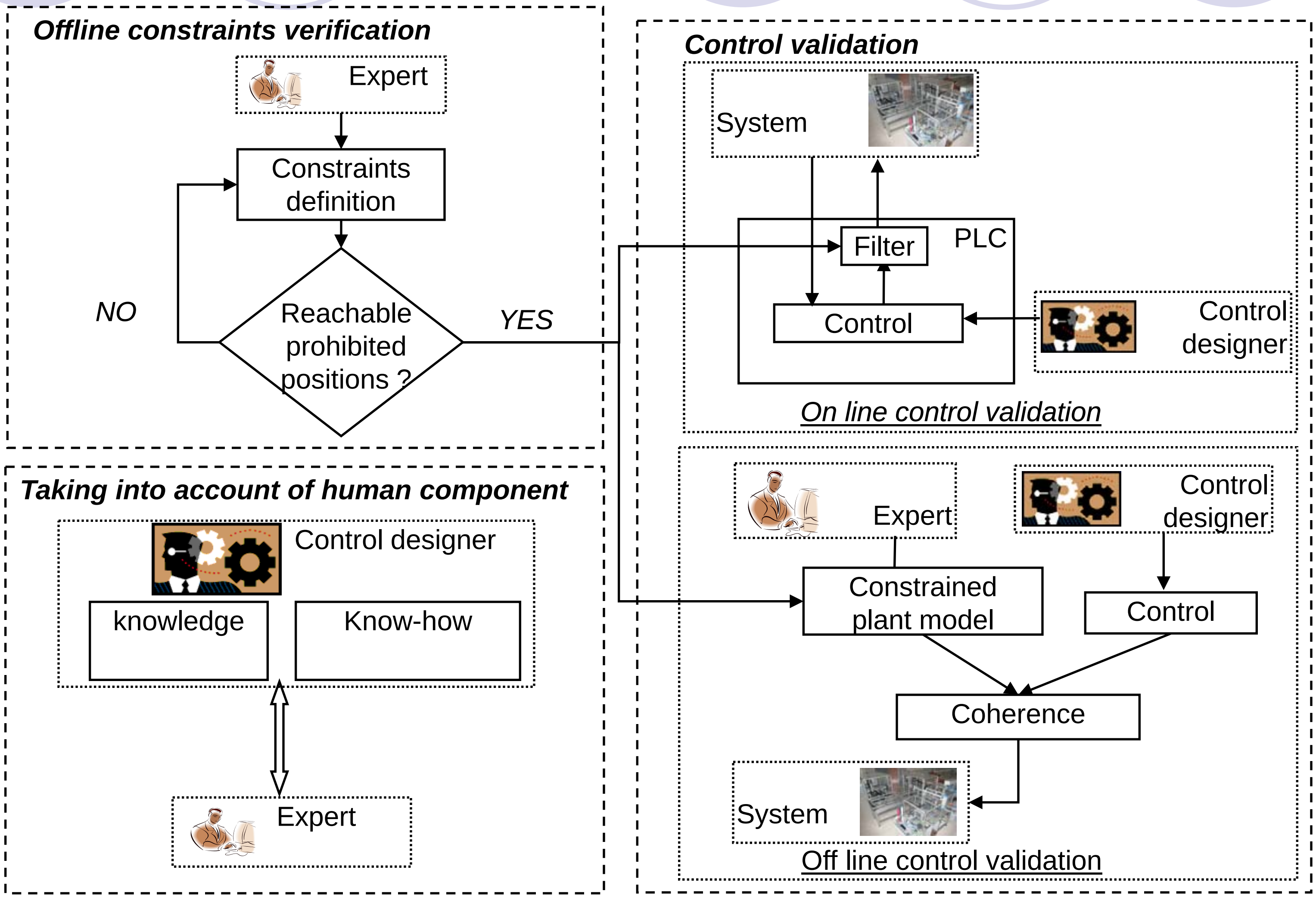
Introduction
Works in the domain - in CReSTIC

- Plant model definition and control designer
  - Global plant model with Boolean automata at level actuator/sensor, doesn't consider the control designer, valves, temporal aspects and simultaneous evolutions

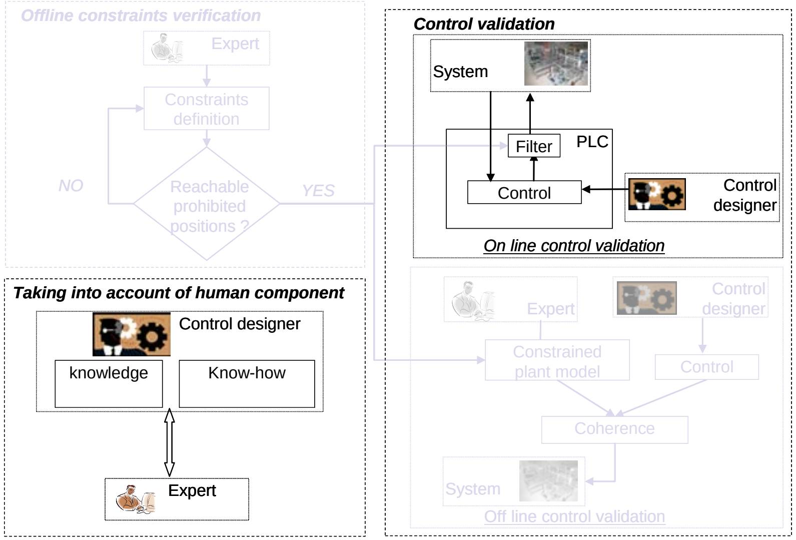
- Explanation generation in the case of errors
  - Verification phase by the expert :
    - The detected error depends of the control
    - The constraints sufficiency isn't verified
  - Validation phase by the control designer :
    - Detection of bad condition transition, bad outputs
    - The analysis requires to be made by an expert

- Approach reliability at the level of system safety
  - Use of the supervisory control theory allows to assure a high reliability
  - Heavy step in models design and combinatory explosion

Introduction
Contribution of thesis works



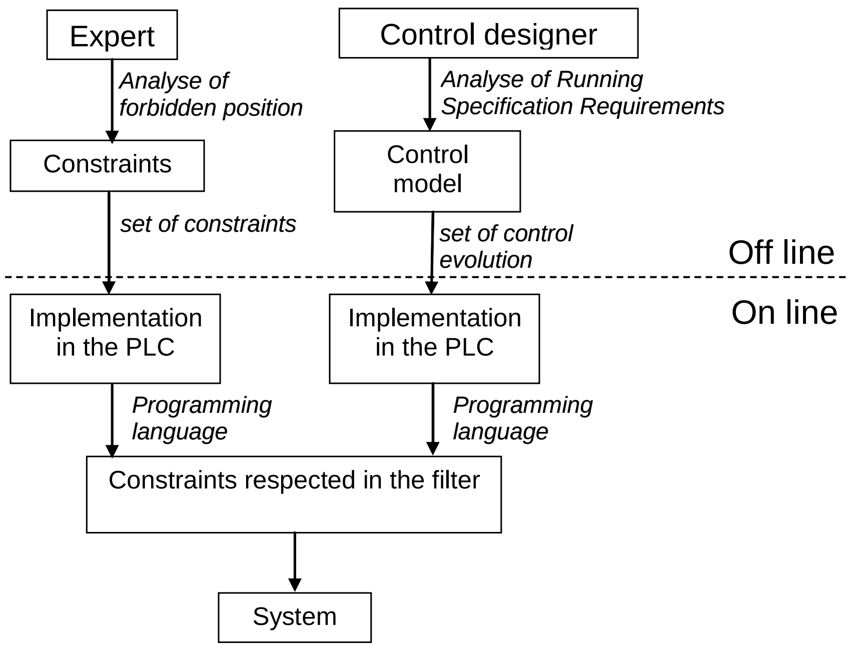
Plan



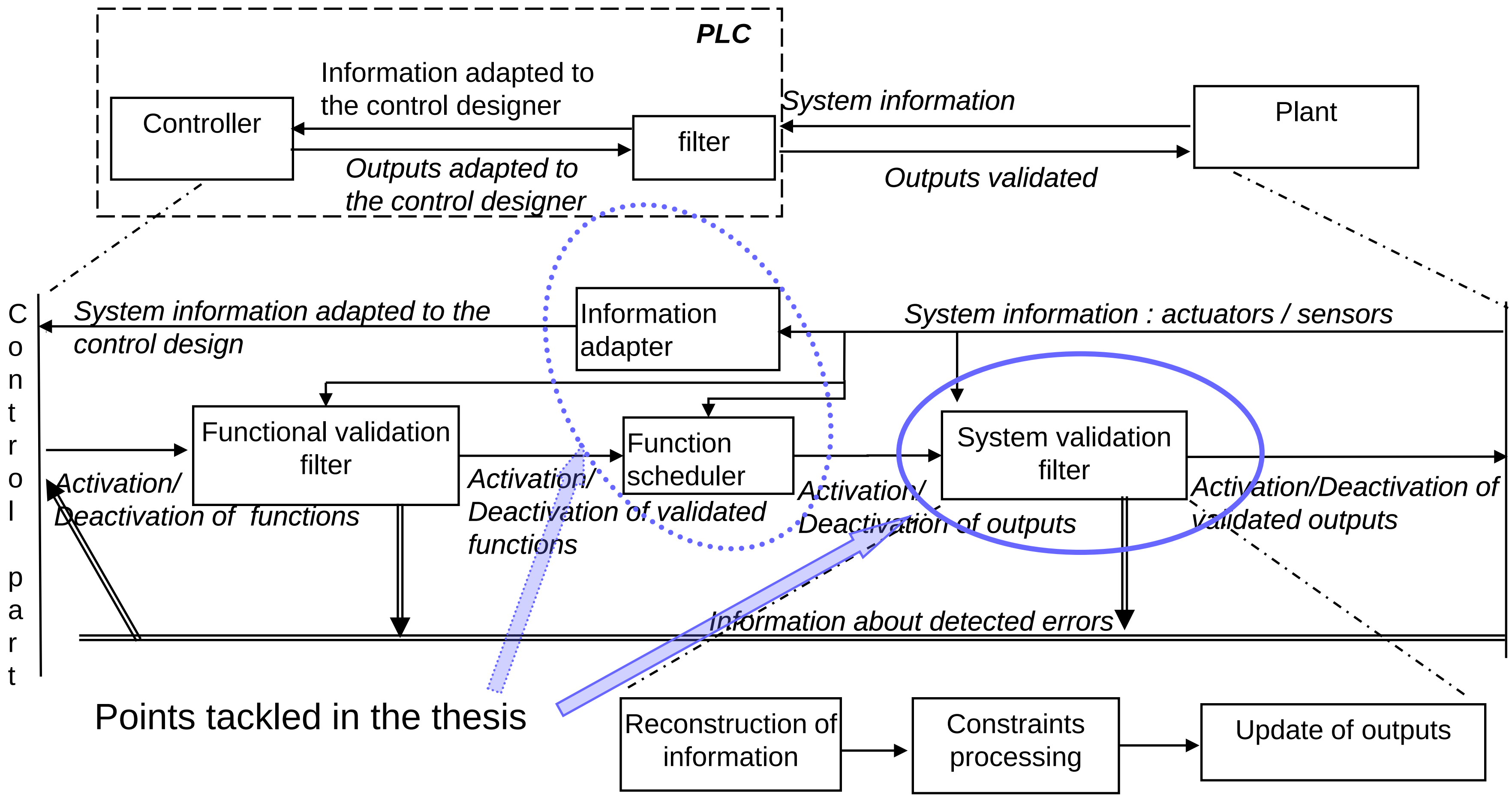
Offline constraints verification
Expert
Constraints definition
Reachable prohibited positions ?
NO
YES
Control validation
System
Filter
PLC
Control
Control designer
On line control validation
Expert
Control designer
Constrained plant model
Control
Coherence
System
Off line control validation
Taking into account of human component
Control designer
knowledge
Know-how
Expert

Control validation approach by filter
Definition of filter model

- At each PLC cycle, the control must validate all constraints before turning on outputs

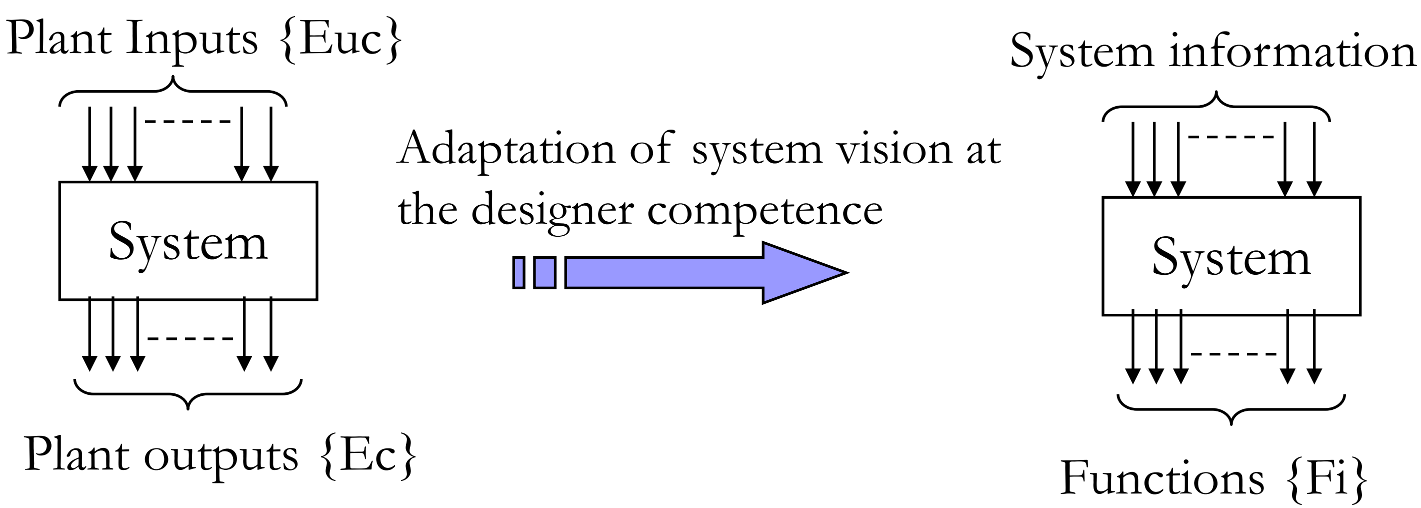


Control validation approach by filter
Definition of filter model



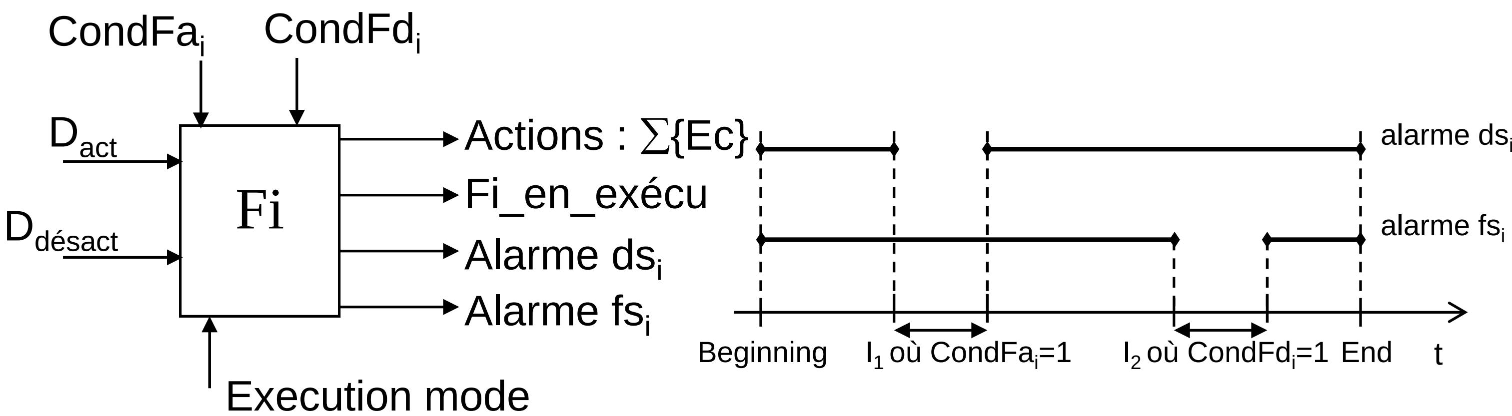
Control validation approach by filter
System adaptation

- Pedagogical case : more motivating working on a real system
- Encapsulation of non adapted control parts
- Adaptation of system vision by functional analyses



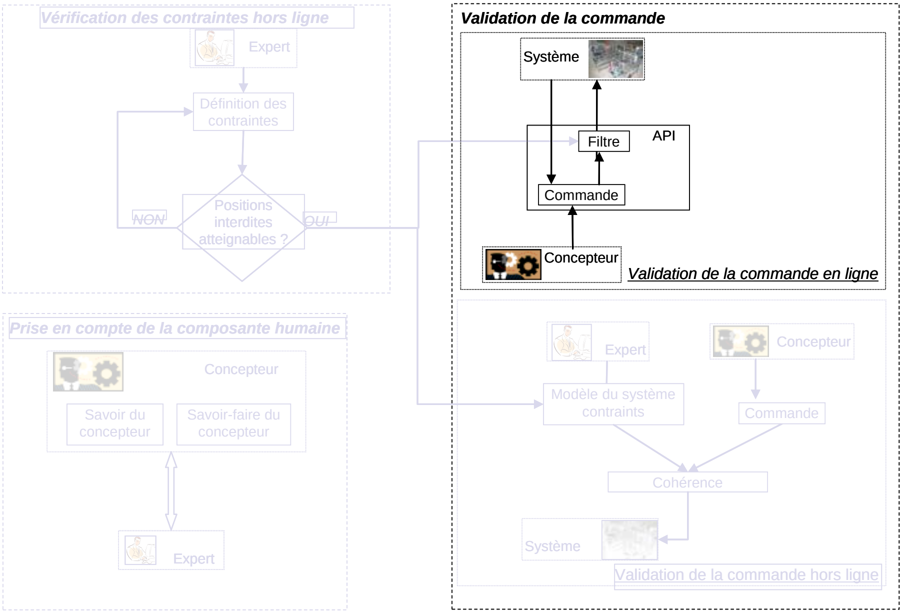
Control validation approach by filter

System adaptation – Function definition



- Designer possibility : to activate or to deactivate a function
- For a great function execution, the activation or deactivation conditions must be respected if not an alarm is set on
- Two execution mode :
  - Semi-automatic mode : control designer manages only the function activation
  - Controlled mode : Control designer must manage the activation and the deactivation of a function, when the conditions become true.
- Hypothesis : the function can't be reactivated on the way

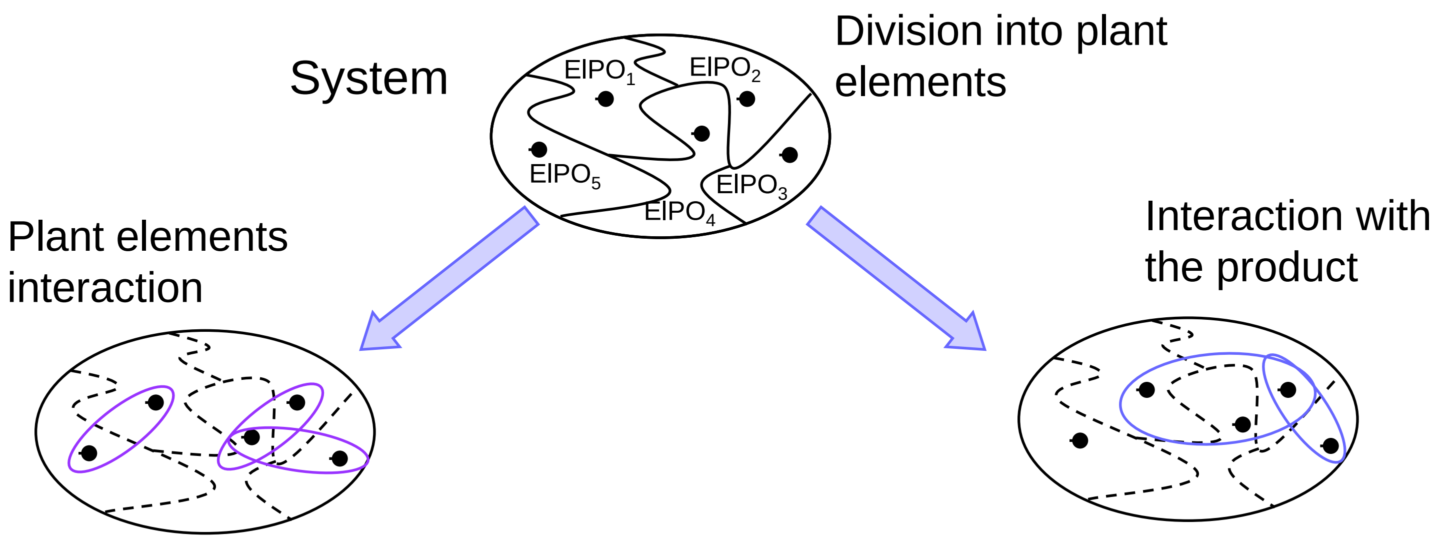
Control validation approach by filter
Method to obtain the safety constraints



- Control validation during the PLC program execution
- Filter is placed between the plant and the controller
- Definition of logical constraints

Control validation approach by filter
Method to obtain the safety constraints

- Some constraints defined by logical equations to safe the system
- Method to define the constraints
  1. Divide the system into plant elements (ElPO)
  2. Definition of plant elements interaction and interaction with the product by structural analysis

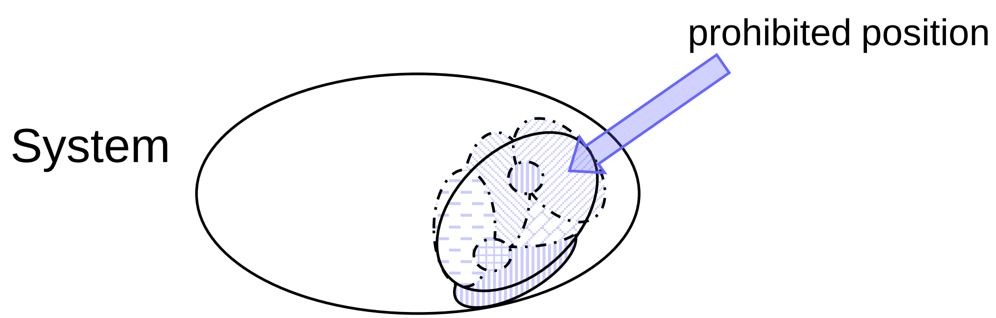


  3. Definition of constraints for each “bad” situation

Control validation approach by filter
Method to obtain the safety constraints

3. Definition of constraints for each situation :
   - ElPO independently : allowing to take into account the designer's errors,
   - ElPO in interaction : avoid the collisions between ElPO,
   - ElPO in interaction with the product : avoid the collisions between the product and ElPO

Use of constraints :



The set of constraints can be more constrained than the prohibited position but it ensures the system safety independently of control

Control validation approach by filter

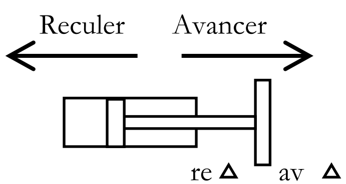
Method to obtain the safety constraints – Framework (1/3)

- Static Safety Constraints : represent a technical or physical impossibilities :

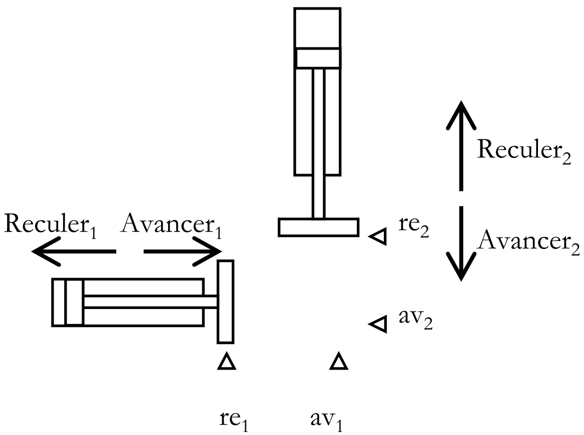
$$Xc_i \wedge Xc_j = 0$$

Generalization in the case of EIPO interaction :

$$f(Xuc_k) \wedge Xc_i \wedge Xc_j = 0$$



Avancer ∧ Reculer= 0



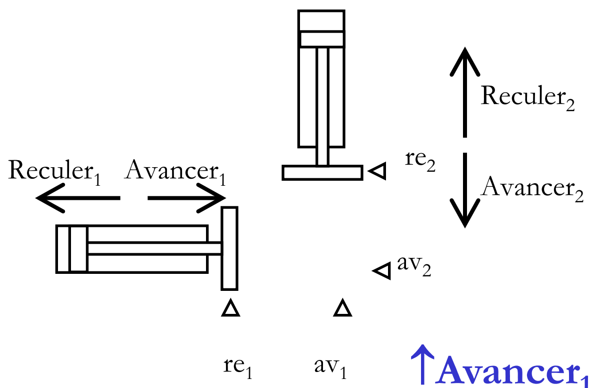
$Avancer_1 \wedge Avancer_2 = 0$

Control validation approach by filter
Method to obtain the safety constraints – Framework (2/3)

- Dynamic Safety Constraints: occurrence of an event which is not compatible with a system situation
  - The request of output activation or deactivation when the activation or deactivation conditions aren't true :

$\uparrow Ec_i \wedge \neg Conda_j = 0$

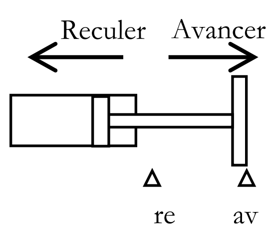
$\downarrow Ec_i \wedge \neg Condd_j = 0$



$\uparrow \mathbf{Avancer_1} \wedge \neg \mathbf{re_2} = 0$

  - The request of output activation when the deactivation conditions are true:

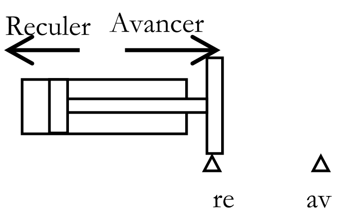
$\uparrow Ec_i \wedge Condd_j = 0$



$\uparrow \mathbf{Avancer} \wedge \mathbf{av} = 0$

Control validation approach by filter
Method to obtain the safety constraints – Framework (3/3)

- Dynamic Safety Constraints: occurrence of an event which is not compatible with a system situation
  - The occurrence of deactivation conditions in relation to the output :
    $\uparrow Condd_j \wedge Xc_i = 0$



$Avancer \wedge \uparrow av = 0$

Control validation approach by filter
Method to obtain the safety constraints – Explanations

- Automatic explanation generation
  - Static safety constraint :

    Interdiction to send at the same time two outputs

  - Dynamic safety constraint :
    - The request of output activation or deactivation when the activation or deactivation conditions aren't true :

      Interdiction to send this output if the conditions aren't true

    - The request of output activation when the deactivation condition are true :

      The outputs activation does not have any effects on the system

    - The occurrence of deactivation conditions in relation to the output :

      The outputs must be deactivated when the deactivation conditions are true

Some constraints can be set but will lose their explanatory power

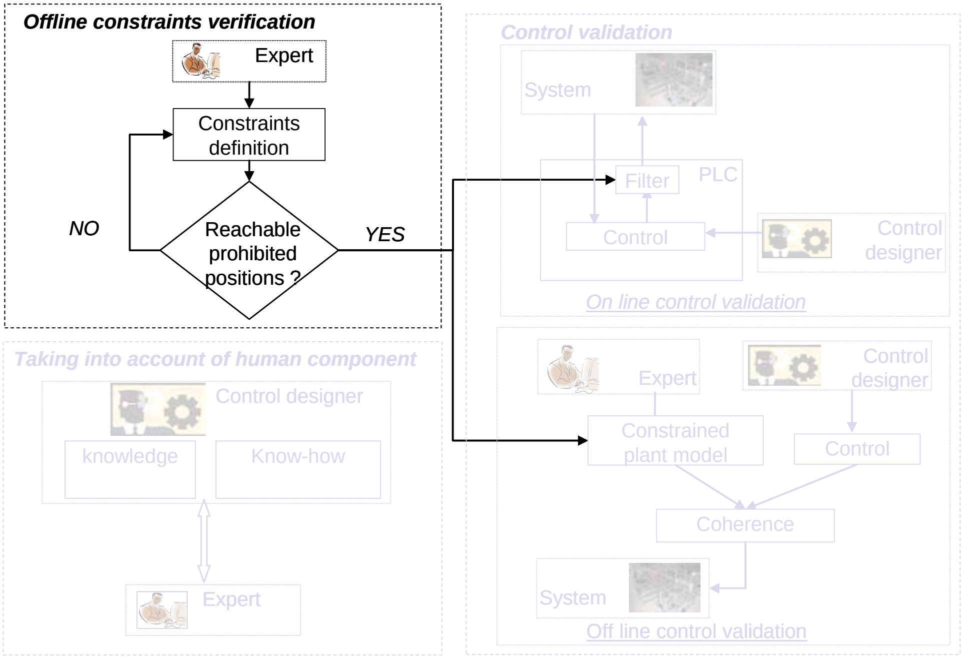
Control validation approach by filter
Discussion

- Taking into account the control designer :
  - The expert defines functions to adapt the system to the control designer
  - Functional analysis of system
  - Encapsulation of programming part non adapted and execution mode : semi-automatic or controlled

- Explanations generation in the case of error :
  - Constraints definition by logical equations linked to an explanation
  - The explanations aren't taken into account designer knowledge

- Approach reliability at the level of system safety :
  - Structural analysis of system
  - Constraints definition is a difficult task. The proposed framework isn't a formal method

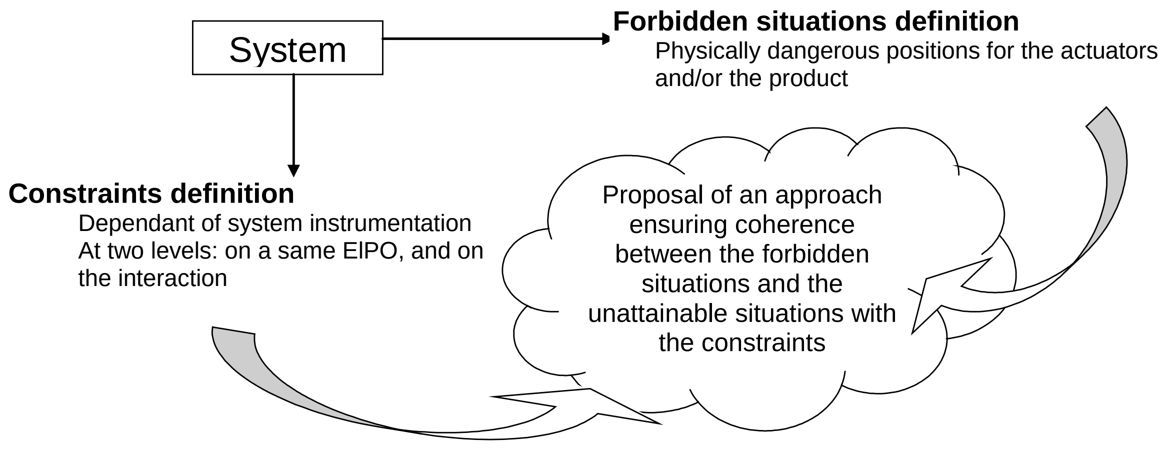
Plan



Offline constraints verification
Expert
Constraints definition
Reachable prohibited positions ?
NO
YES
Control validation
System
Filter
PLC
Control
Control designer
On line control validation
Taking into account of human component
Control designer
knowledge
Know-how
Expert
Expert
Control designer
Constrained plant model
Control
Coherence
System
Off line control validation

Safety constraints verification
Problematic

- The expert defines on the one hand the prohibited situations and on the other hand the constraints



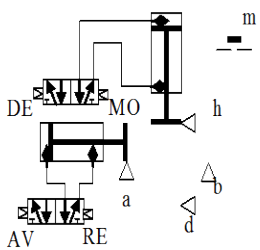
- Are we sure that the constraints are sufficient to avoid dangerous positions?

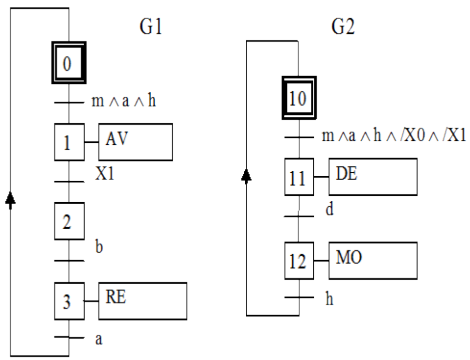
- Constraints verification for an interaction between EIPO and with the product

Safety constraints verification

Problematic – Execution on a real system

- Taking into account the PLC cycle time
  - According to PLC cycle time, it is possible to have a collision between EIPO





Constraints :

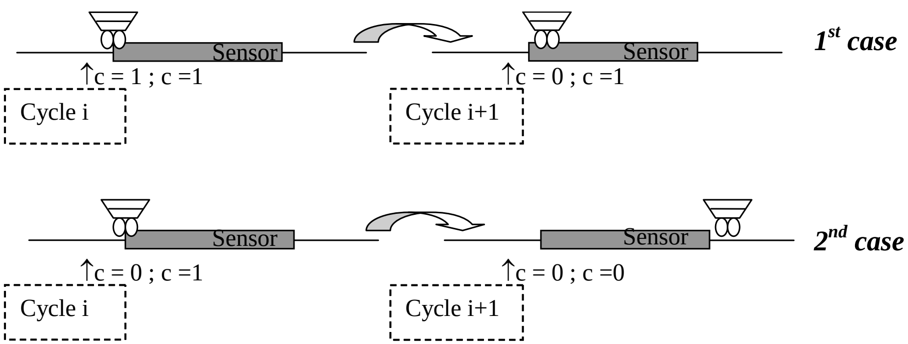
$Av\_auto \wedge \neg h\_auto = 0$

$De\_auto \wedge \neg a\_auto = 0$

$Av\_auto \wedge De\_auto = 0$

These constraints aren't sufficient

- Presence of inertia in the system



The causality delays and the inertia effects must be taken into account in the verification

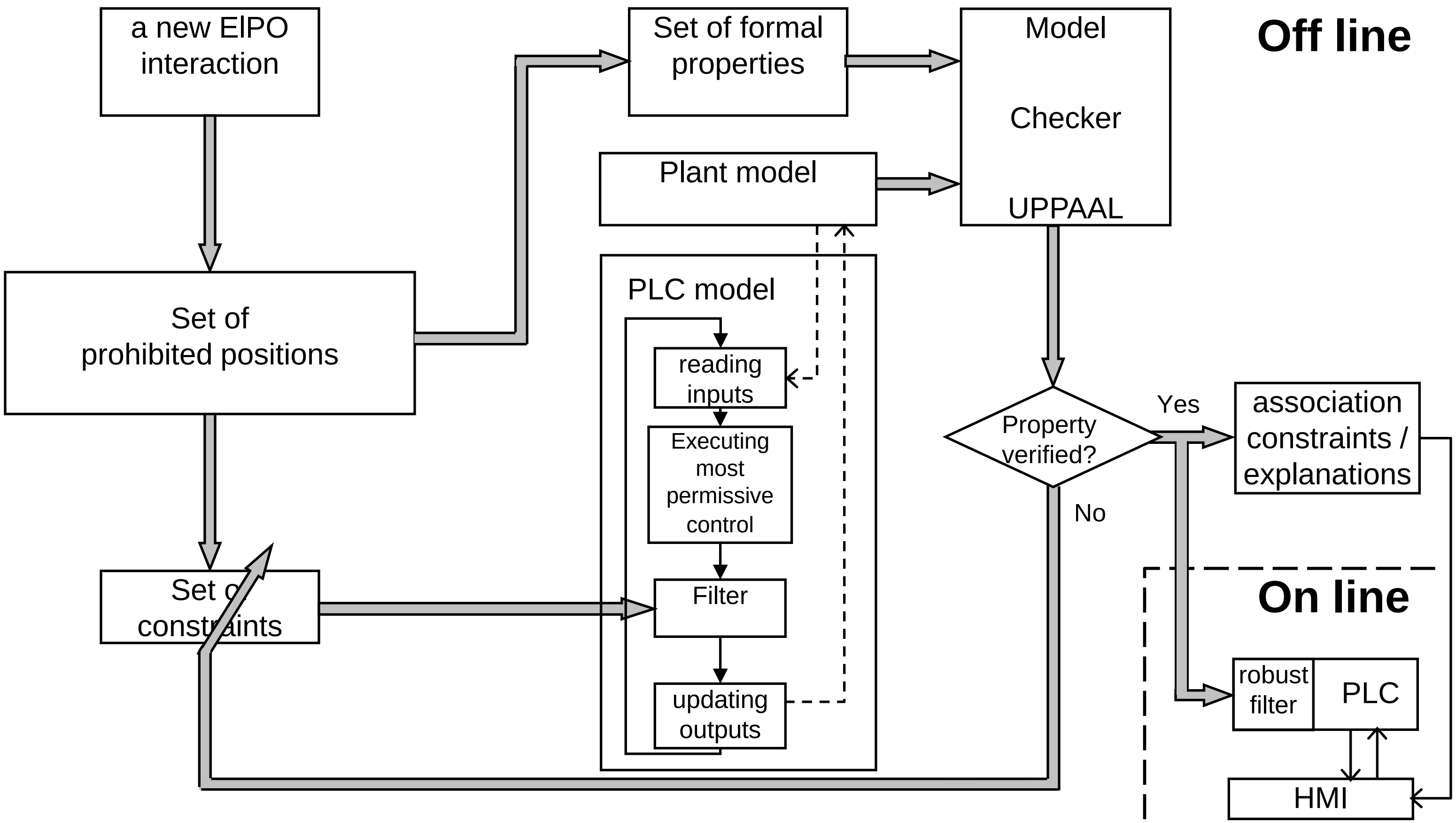
Hypothesis : "long" contact sensors

Safety constraints verification
Verification approach

- Use the same steps that for the constraints definition
- The constraints set of each interaction is verified independently from each other
- Modeling only EIPO concerned in the interaction
- Step:
  1. Regroup of the constraints by interaction
  2. Seek in the library, if the constraints set has already been verified
  3. Verify the constraints sufficiency
  4. Verify the constraints requirement
  5. Update the constraints library

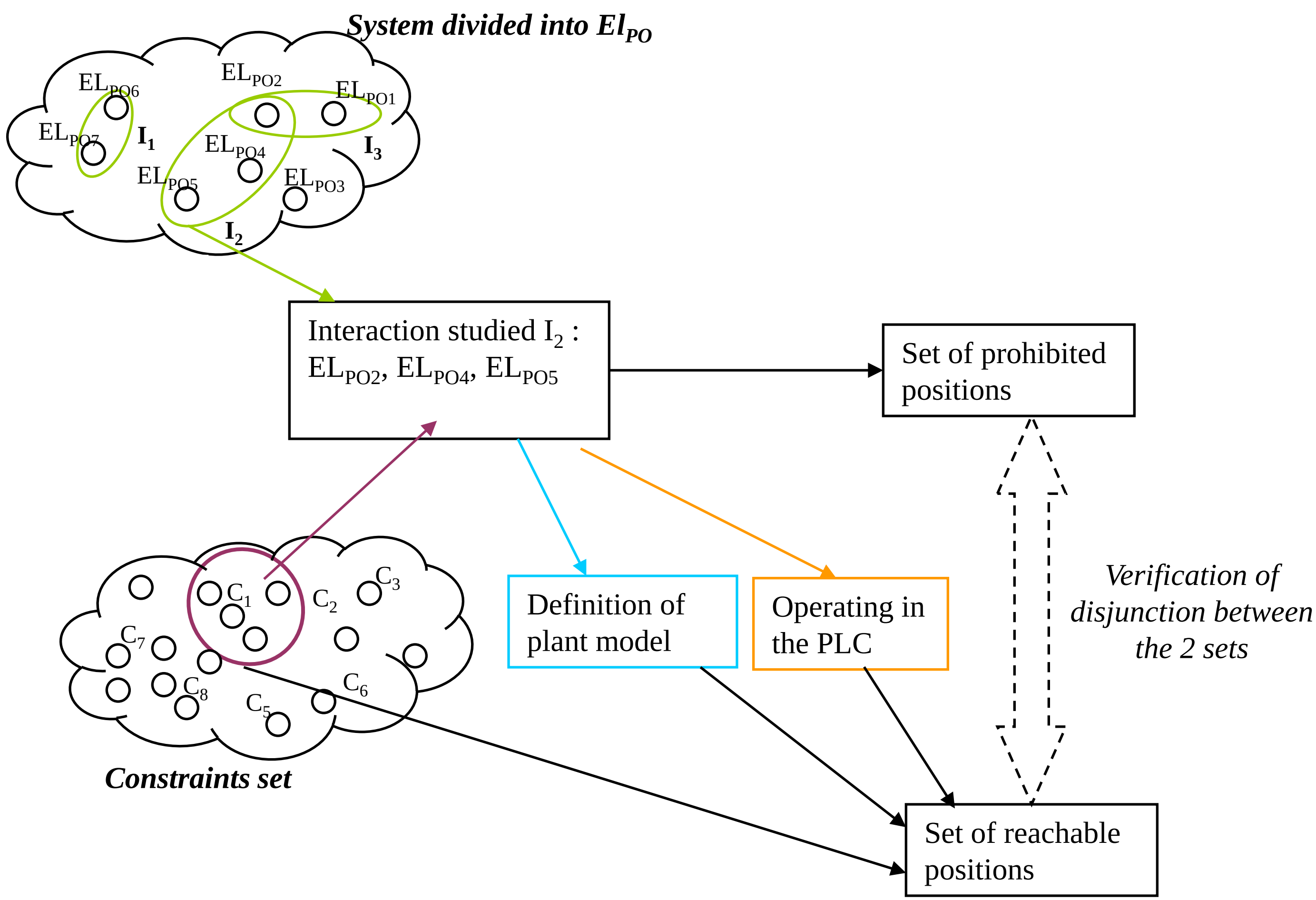
Safety constraints verification

Verification approach - Steps



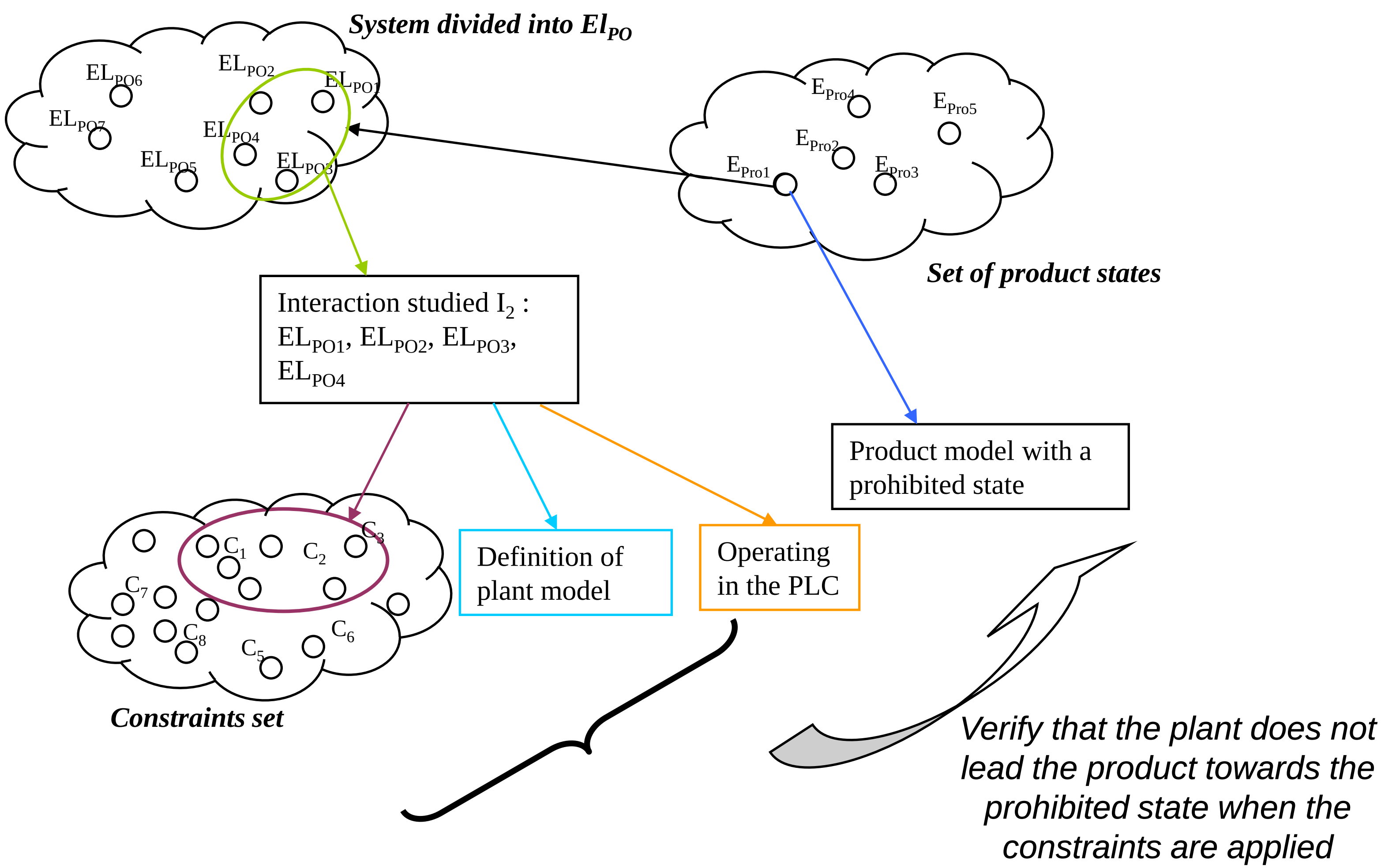
Safety constraints verification

Verification approach – Interaction between ElPO



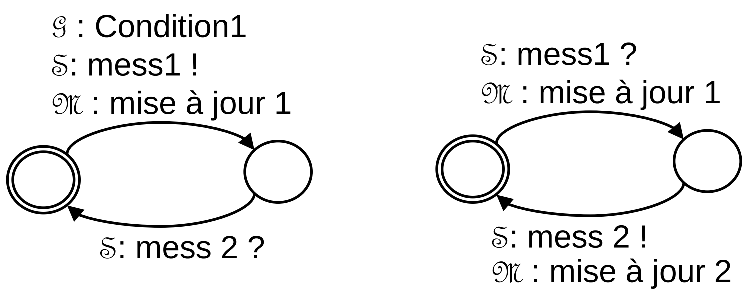
Safety constraints verification

Verification approach – Interaction with the product



Safety constraints verification
System model – Works context

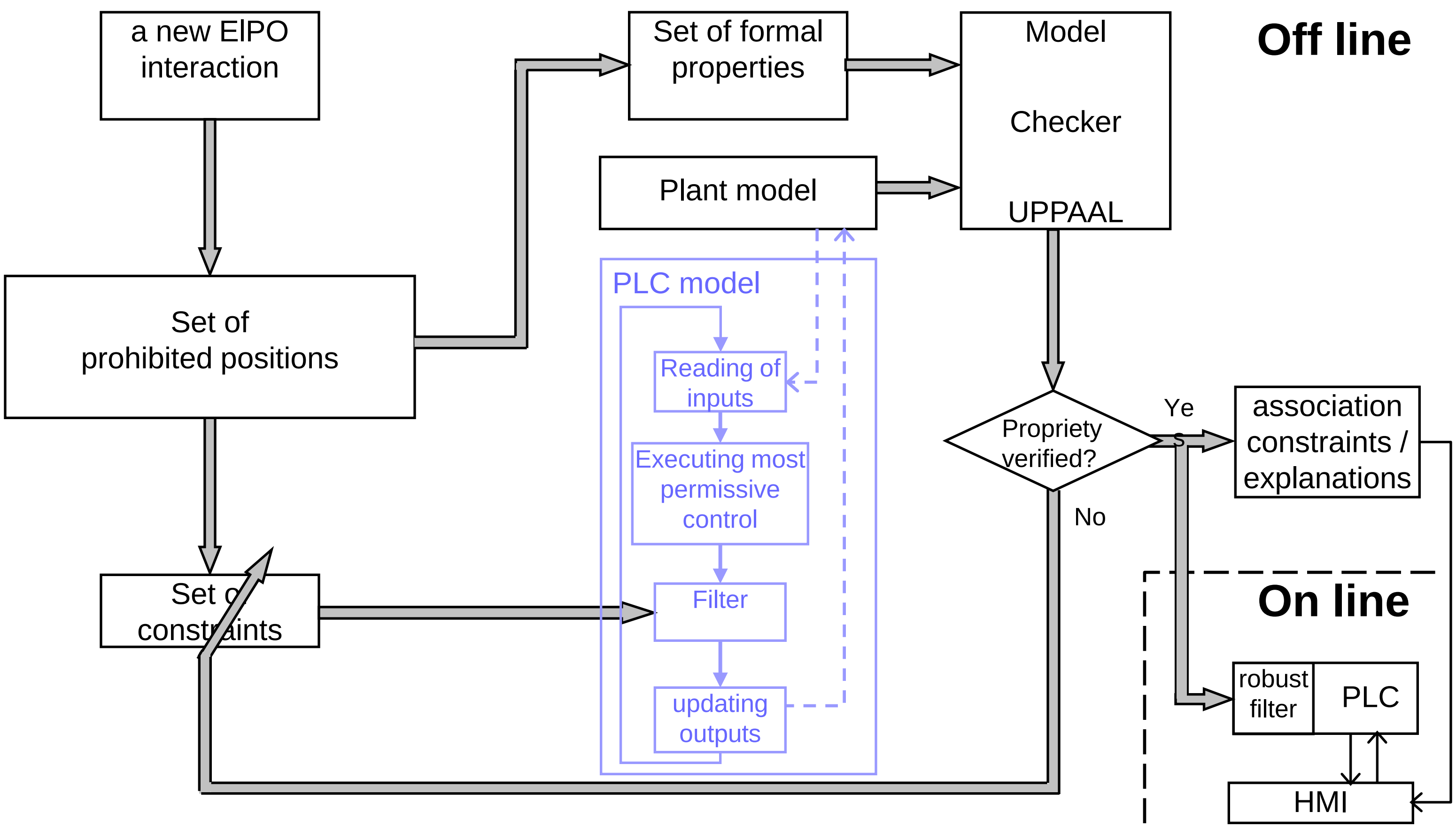
- The system {Plant, controller, Product, computing environment} is modeled in modular way for each plant element and each product
- Communicating automaton (Julliand, 2003)



- The modeling is performed with a PLC point of view. So, we consider the cyclic operating and response time that leads to simultaneous evolution at the plant and controller levels
- Work hypotheses
  - All evolution are observable by the PLC
  - Possibilities of simultaneous evolution E_c, E_{uc} for the PLC,
  - Causality time delay, taking into account inertia,
  - Verification in the worth case: Output updating has an effect on the system

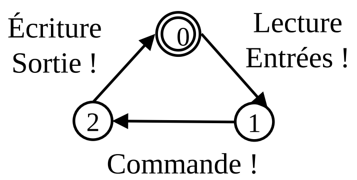
Safety constraints verification

System model

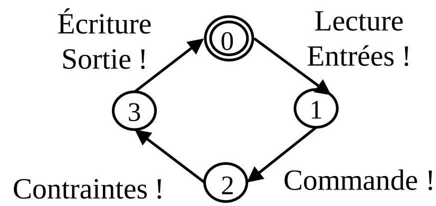


Safety constraints verification
System model – Environment model

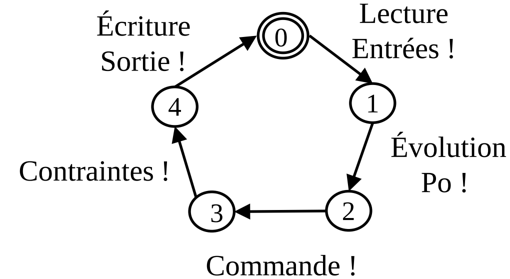
Classical PLC operation



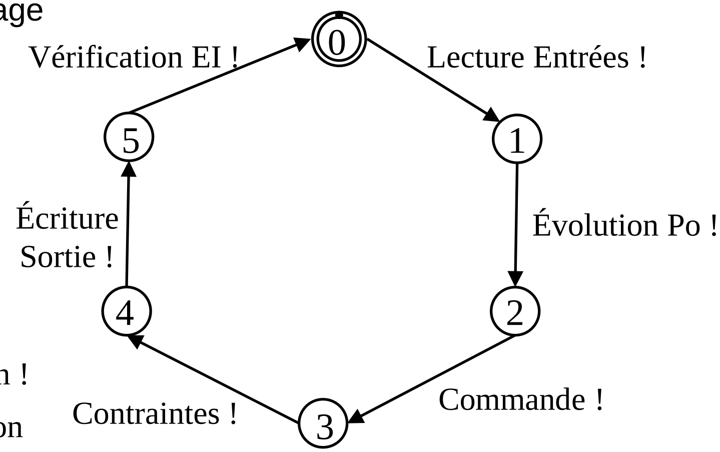
PLC operation taking into account the filter

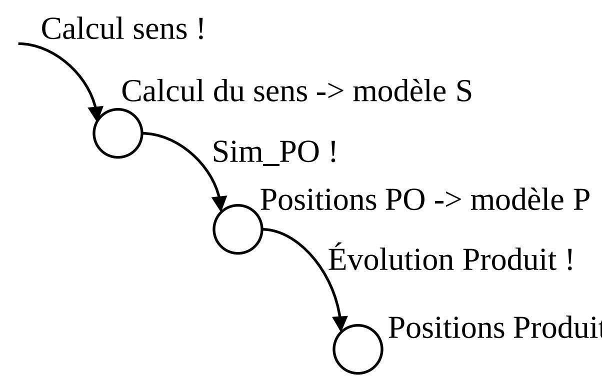


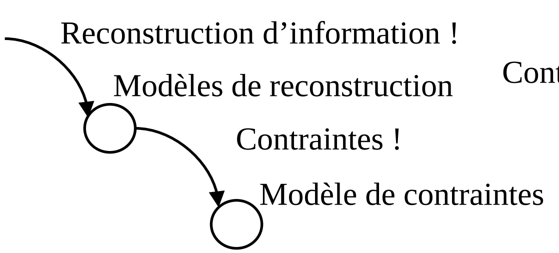
PLC operation managing the plant evolution



PLC operation with the verification stage







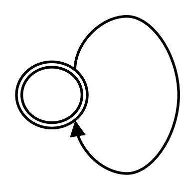
Safety constraints verification
System model – Environment model

- Variables used in the different models

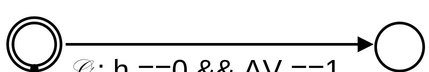
		PLC variables	System variables
Reading model		L	
Plant model	Direction model		Sens
	Positions model		Pos
	Product model		Pro
Outputs model		C	
Model of constraints evolution	Reconstruction model	R	
	Constraints model	Cont	
Writing model			S
Verification model			V

- Reading, Updating, Constraints models

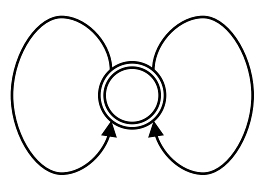
Reading model



Constraints model



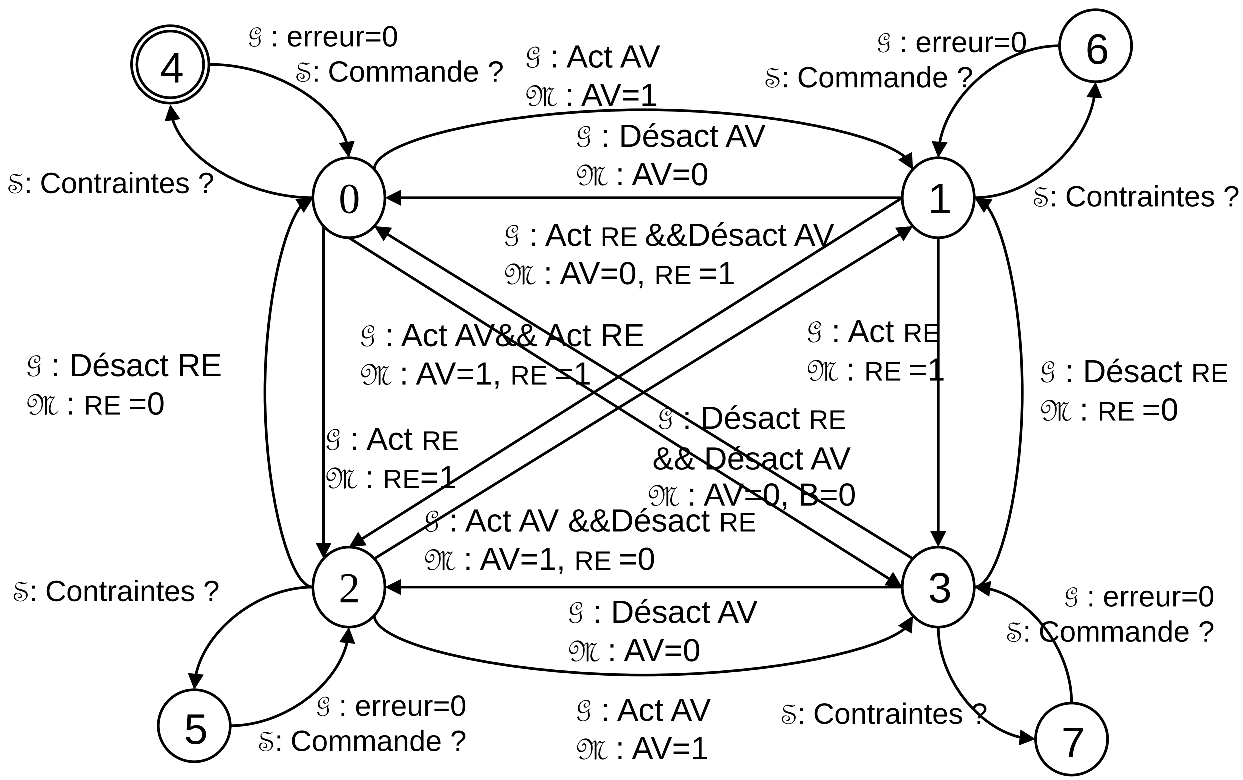
Updating model



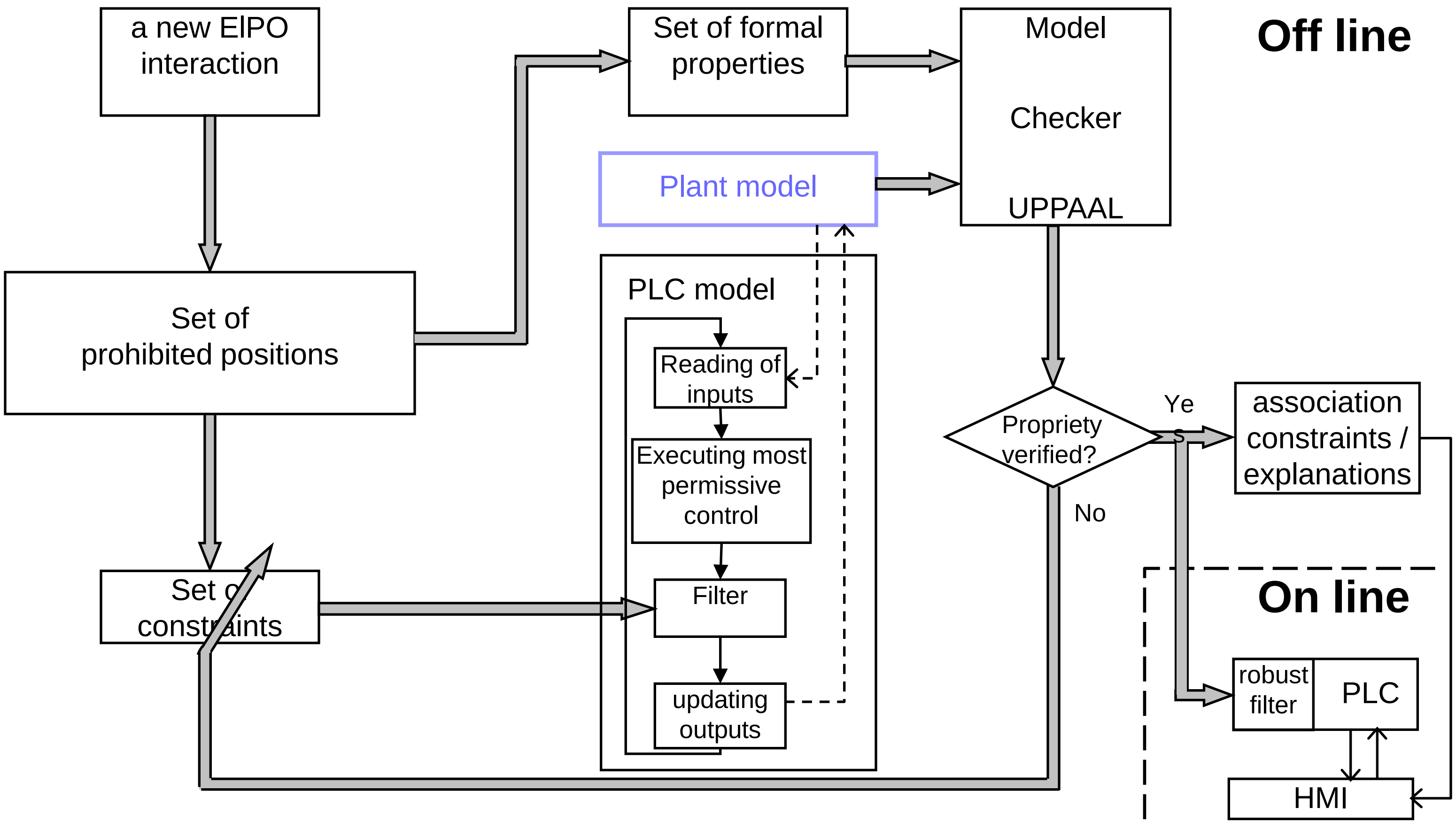
Safety constraints verification

System model – Outputs model

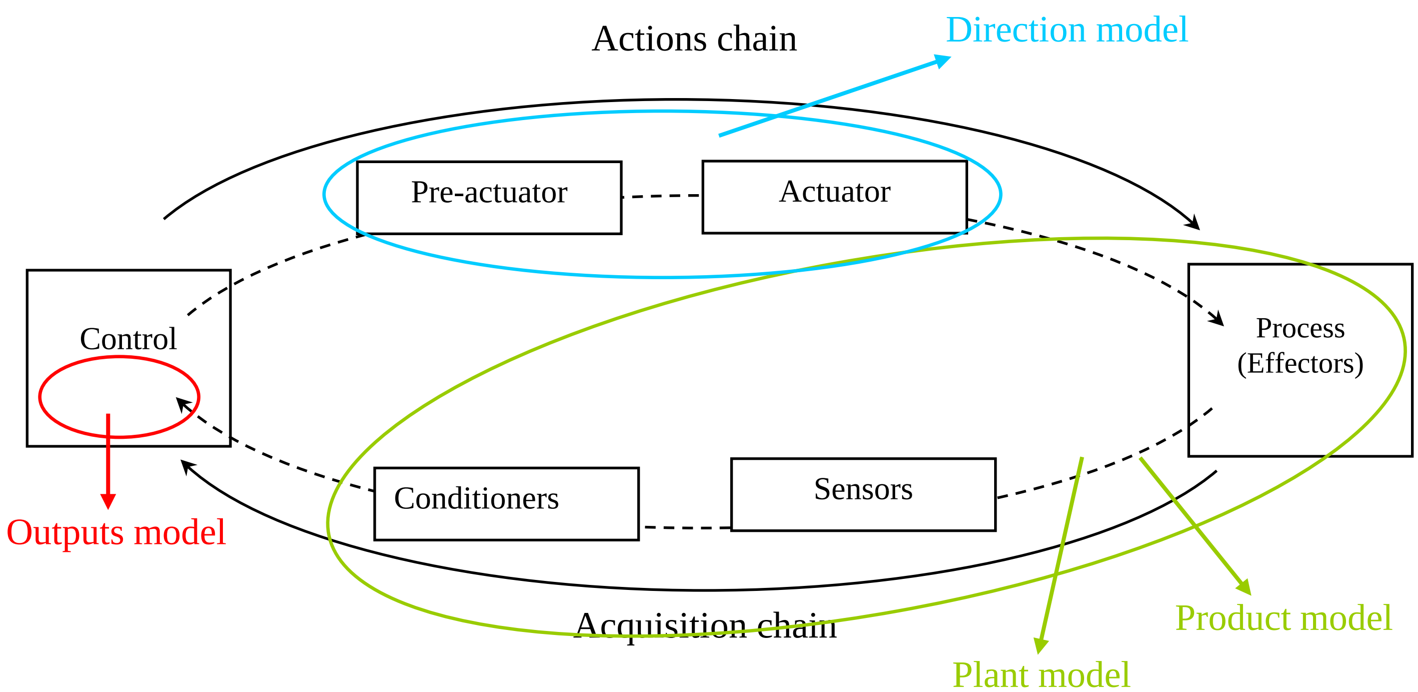
- Represent the most permissive control of EIPO
- Case of a double effect cylinder (*AV*, RE) piloted by a valve 5/2 with 2 sensors (a, r)



Safety constraints verification
System model



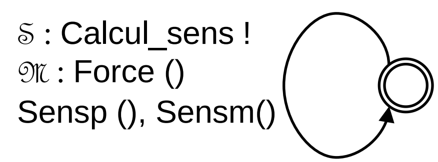
Safety constraints verification
System model – Functional chain



Safety constraints verification
System model – Direction model

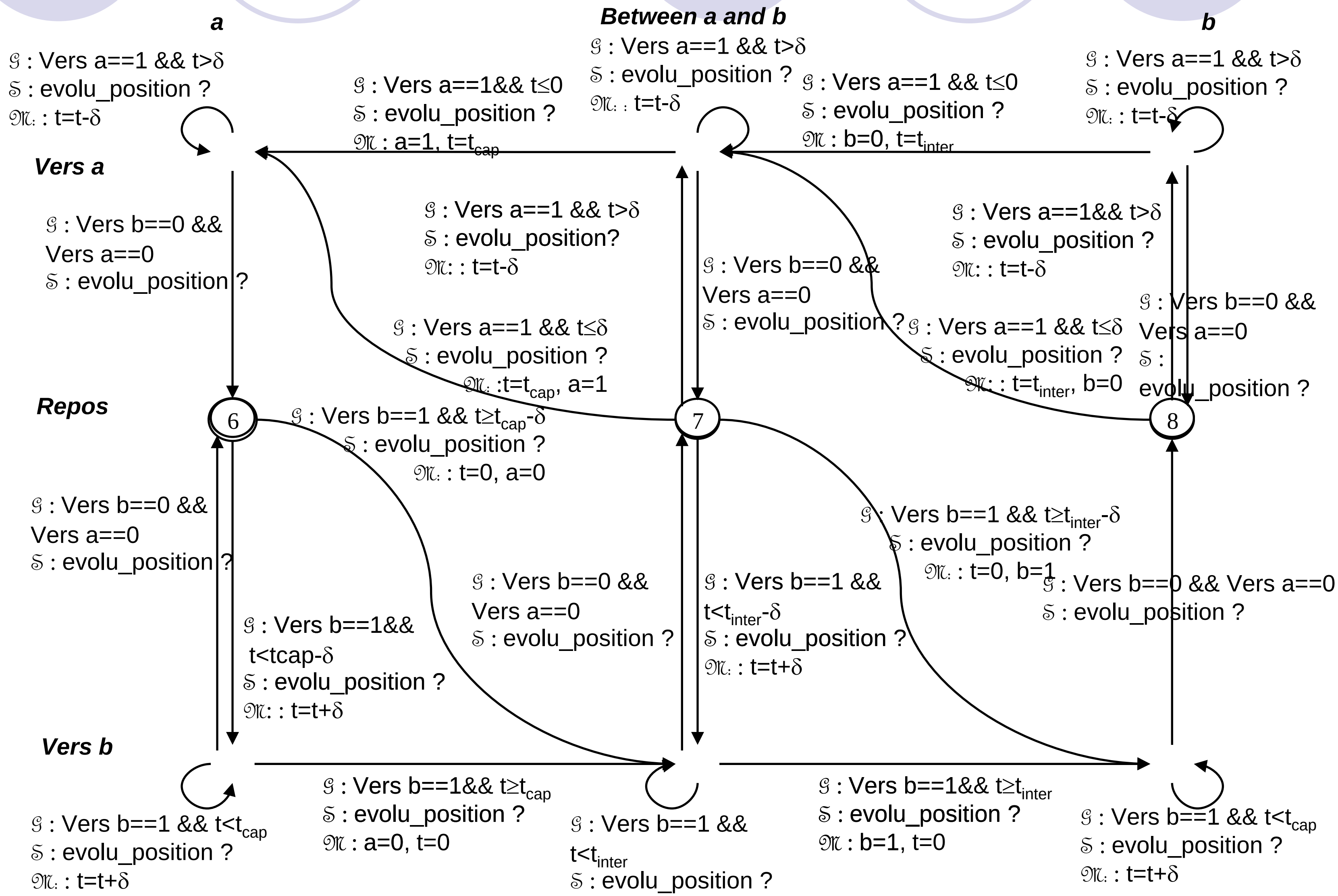
- To take into account the ElPO technology
- To model the inertia effects and causality delay

```
int Force()
{
if((A&&!B)==1) {F=F+f;}
if((!A&&B)==1) {F=F-f;}

if (F>fmax){F=fmax;}
if (F<-fmax){F=-fmax;}
return(F);
}

bool sensp()
{
    Sp=(F≥frp)?1:0;
    return(Sp);
}

bool sensm()
{
    Sm=(F≤frm)?1:0;
    return(Sm);
}
```



Safety constraints verification

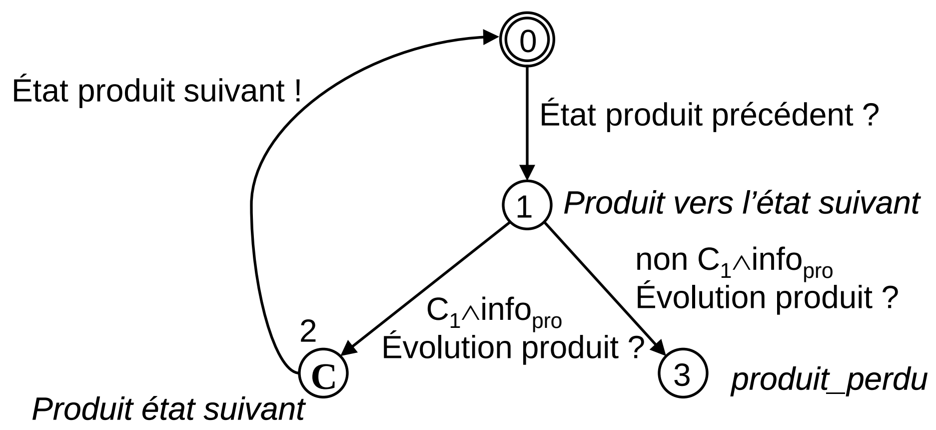
System model – Positions model



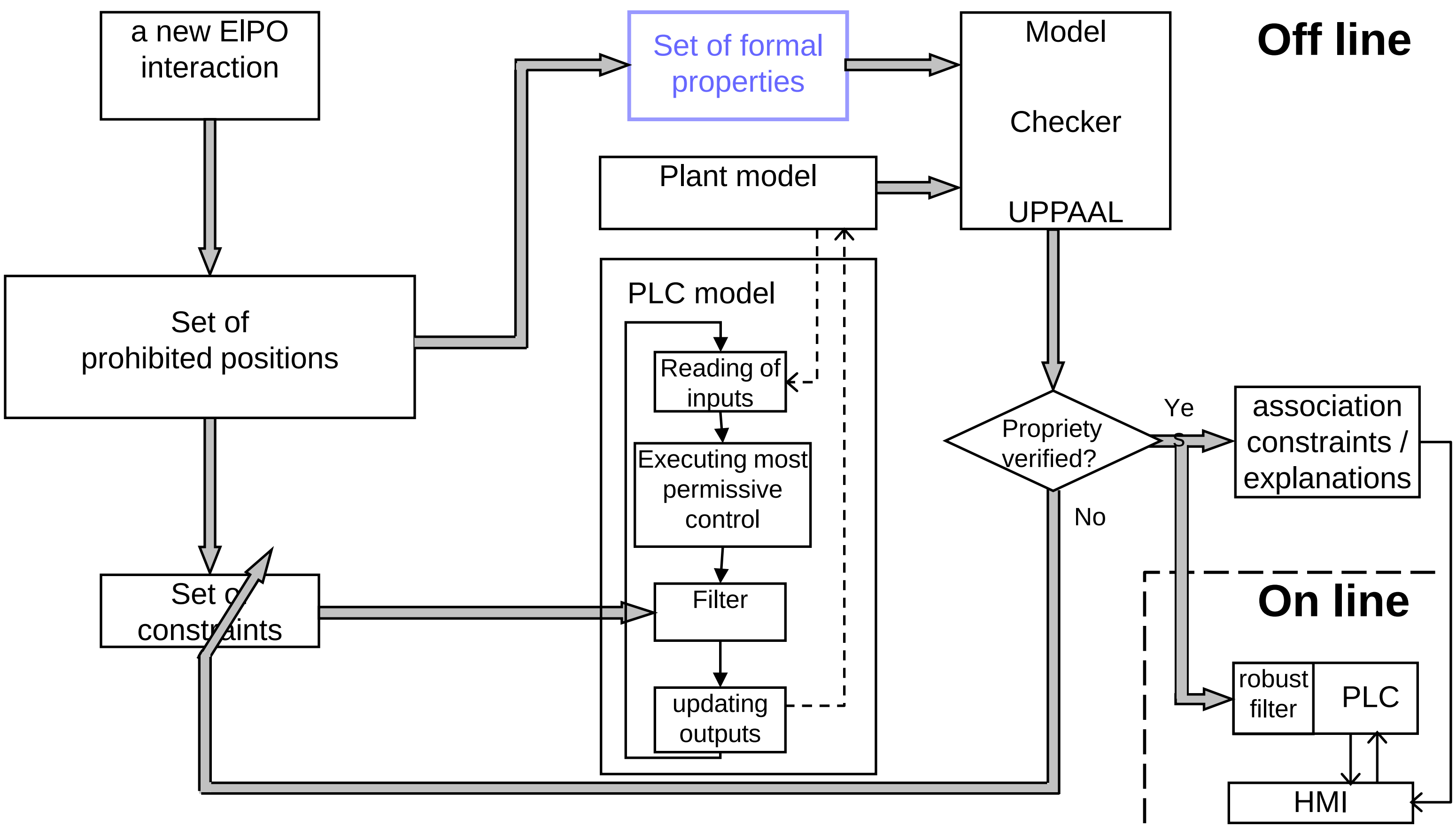
Safety constraints verification
System model – Product model

- To model the product evolution according to the plant positions
- Product evolution

- Product model



Safety constraints verification
System model

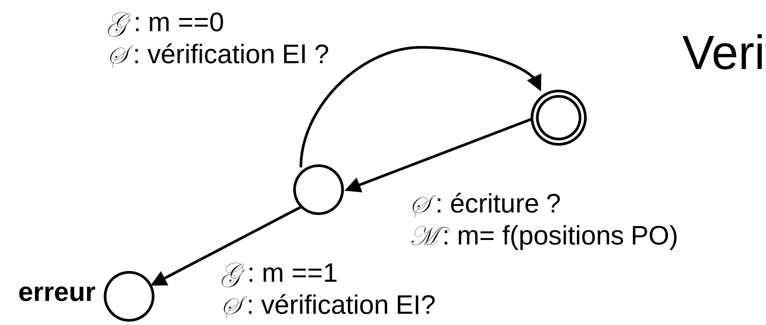


Safety constraints verification

System model – Verification model

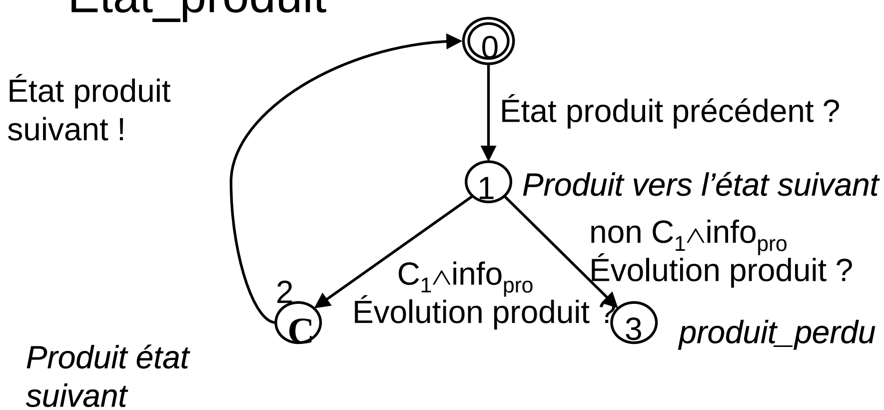
- For an interaction between EIPO :

P1 : A[] not(Verif.erreur)



- For an interaction with the product :

Etat_produit



P2 : A[] not(Etat_produit.produit_perdu)

The definition of the formal properties is simple because there are few EIPO considered in interaction

Safety constraints verification
System model – Animation

Constraints

AV_auto ∧ ¬h_auto = 0

DE_auto ∧ ¬a_auto = 0

AV_auto ∧ DE_auto = 0

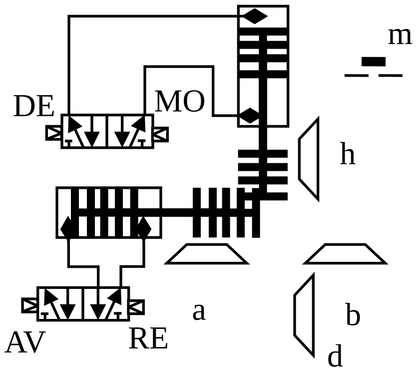
↓AV_auto ∧ a_auto = 0

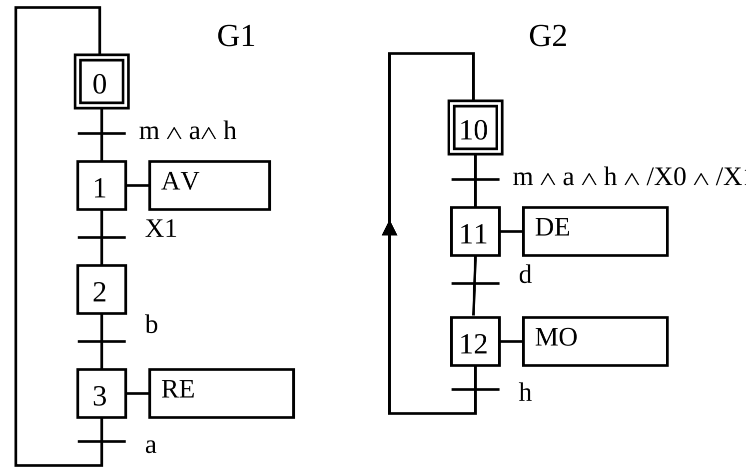
↓DE_auto ∧ h_auto = 0

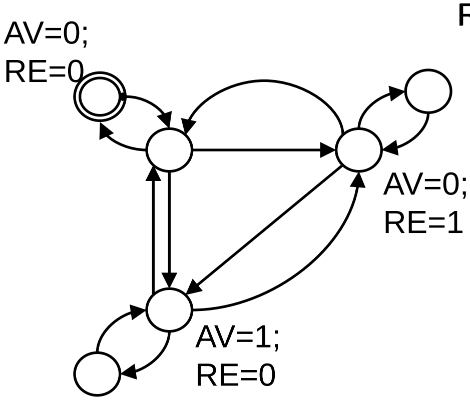
Or

↓AV_auto ∧ ¬b_auto = 0

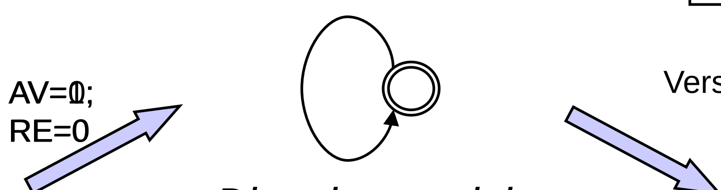
↓DE_auto ∧ ¬d_auto = 0



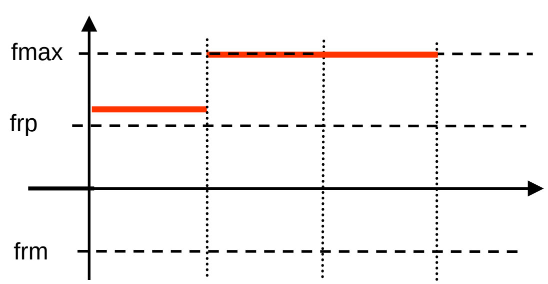


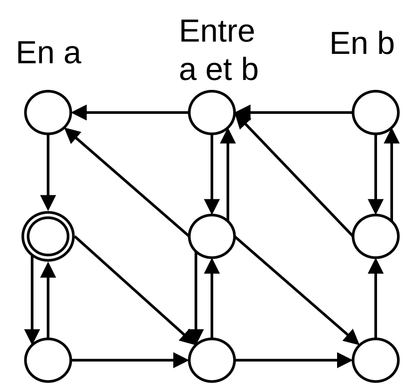


Outputs model



Direction model





Position model

Safety constraints verification
Discussion

- Verification approach
  - Formal verification of constraints
  - Using of model checker (UPPAAL)

- Plant model
  - The outputs, direction and positions models proposed are generic,
  - The change of EIPO technology modifies only the direction model
  - Taking into account the product presence doesn't increase the complexity of the plant model.
  - Definition of the product models can be a difficult task

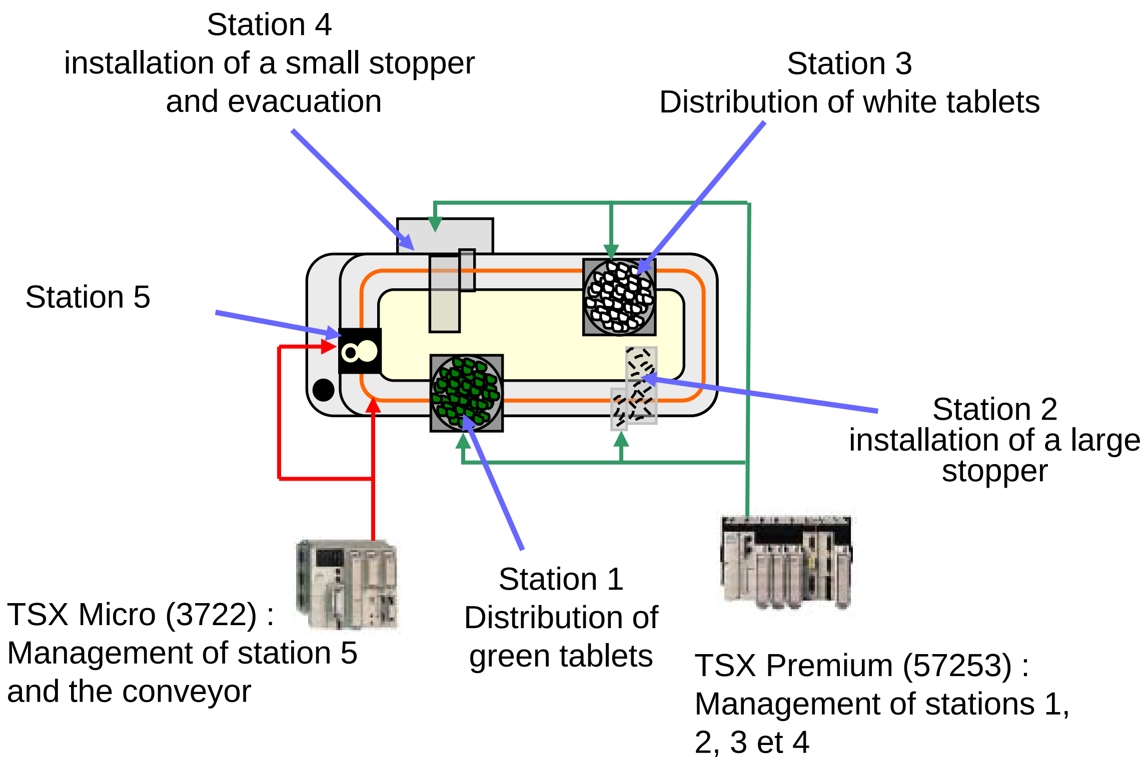
Plan

Pedagogical applications
Context

- To allow to make control a real system by learners
- To safe system
- To adapt the level difficulty

- Real system : Productis machine to illustrate the system adaptability
- Virtual system : ITS PLC to illustrate the verification method

Pedagogical applications
Real system : Productis machine



The system is composed of 46 outputs and 64 inputs

Pedagogical applications
Real system : Productis machine

Outputs numbers / Execution mode

Difficulty

5 functions for each station

Semi

20 functions

Replace functions in chronological order

Semi

18 functions

Programming

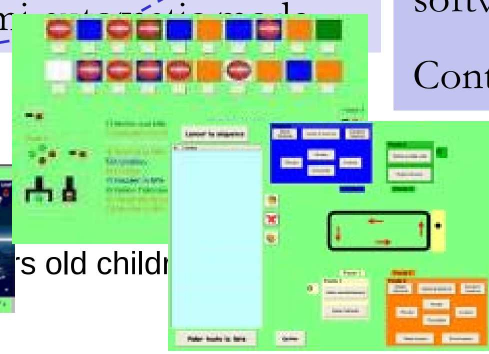
software PL7

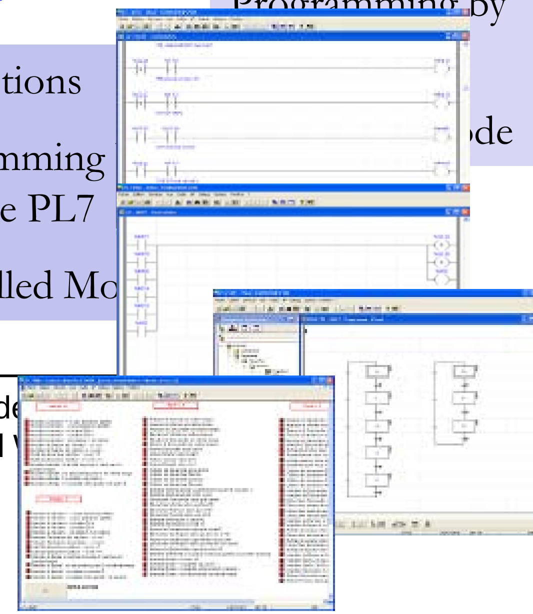
Controlled Mo

46 functions for each outputs

Programming by

de





s old childr

rol de

ical

pert :

Master pro

96 constraints are required to safe the system

Pedagogical applications

Virtual system : Automatic warehouse

	System variables	ITS PLC variables
Transelevator	I0, I1	I0_auto, I1_auto
	O0, O1, O2, O3, O4, O5	O0_auto, O1_auto, O2_auto, O3_auto, O4_auto, O5_auto
Forks	I2, I3, I4, I5	I2_auto, I3_auto, I4_auto, I5_auto
	O6, O7	O6, _auto O7_auto
Entry bay	I6	I6_auto
Exit bay	I7	I7_auto

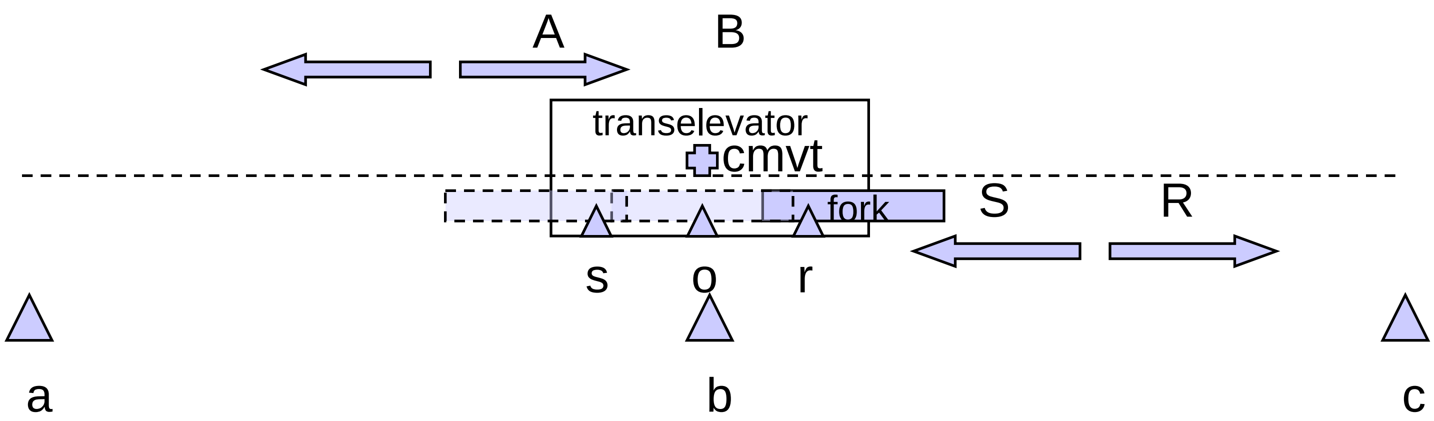
The system is composed of 8 outputs and 8 inputs

This automatic warehouse system is composed of a transelevator, a rack, an entry bay and an exit bay.

The automatic monorail, delivers boxes to the transelevator. The boxes are delivered and retrieved by the forks, followed by an automatic movement of the elevator. The rack is subdivided into 50 cells, which are identified by a number. The monorail retrieves the boxes from the transelevator.

Pedagogical applications

Virtual system : Automatic warehouse – Simplification



- The warehouse is reduced to one dimension
- The position *b* represents the rack position
- The positions *a* and *c* represent the entry and exit position
- The forks go out at the position *s* to take the product in position *a* and to position *c* to discharge
- The forks go out at the position *r* to take or to discharge the product in the stock (position *b*)
- A movement sensor *cmvt* measures the warehouse movement

Pedagogical applications
Virtual system : Automatic warehouse - System analysis

Functions			Actuators	Possible fault
To manage an automatic warehouse	To supply with box	To go to position p_{entry}	A, B (interaction with S and R)	Movement with the forks (PF1) Stopping to intermediate position (PF2)
		To take a product at the entry bay	S (interaction with A and B) (interaction with the product)	already a product in the transelevator (PF3) To go out in the bad way (PF4) Transelevator in movement (PF5)
		To go to desired position p_i	A, B (interaction with S and R)	(PF1, PF2)
		To put the product to the rack	R (interaction with A and B) (interaction with the product)	already a product in the rack (PF6) (PF4, PF5)
	To discharge box	To go to desired position p_i	A, B (interaction with S and R)	(PF1, PF2)
		To put the product to the rack	R (interaction with A and B) interaction with the product)	(PF3, PF4, PF6)
		To go to position p_{exit}	A, B (interaction with S and R)	(PF1, PF2)
		To take a product at the exit bay	S (interaction with A and B)	(PF3, PF4, PF5)

Pedagogical applications

Virtual system : Automatic warehouse – Constraints definition

- PF1 : Movement with the forks
  - $\uparrow A \wedge !o = 0$ (C_1) : prohibited to activate *A* if the forks aren't in position *o*
  - $\uparrow B \wedge !o = 0$ (C_2) : prohibited to activate *B* if the forks aren't in position *o*
  - $\downarrow S \wedge !s = 0$ (C_3) : prohibited to deactivate *S* if the forks aren't in position *s*
  - $\downarrow R \wedge !r = 0$ (C_4) : prohibited to deactivate *R* if the forks aren't position *r*
  - $A \wedge (S \vee R) = 0$ (C_5) : prohibited to activate *A* at the same time that *R or S*
  - $B \wedge (S \vee R) = 0$ (C_6) : prohibited to activate *B* at the same time that *R or S*
- PF2 : Stopping to intermediate position
  - $\downarrow A \wedge cmvt = 0$ (C_7) : prohibited to deactivate *A* if the movement sensor *cmvt* is turned on
  - $\downarrow B \wedge cmvt = 0$ (C_8) : prohibited to deactivate *B* if the movement sensor *cmvt* is turned on
- PF3 : already a product in the transelevator
  - $\uparrow S \wedge a \wedge product\_in\_trans = 0$ (C_9) : prohibited to activate *S* if the transelevator is in position *a* and if a product is already in the transelevator
- PF4 : To go out in the bad way
  - $\uparrow S \wedge !c \vee !a = 0$ (C_{10}) : prohibited to activate *S* if the transelevator isn't in position *a or c*
  - $\uparrow R \wedge !b = 0$ (C_{11}) : prohibited to activate *R* if the transelevator isn't in position *b*
- PF5 : Transelevator in movement
  - $\uparrow S \wedge cmvt = 0$ (C_{12}) : prohibited to deactivate S if the movement sensor cmvt is turned on
  - $\uparrow R \wedge cmvt = 0$ (C_{13}) : prohibited to deactivate S if the movement sensor cmvt is turned on

Pedagogical applications
Virtual system : Automatic warehouse – Constraints verification

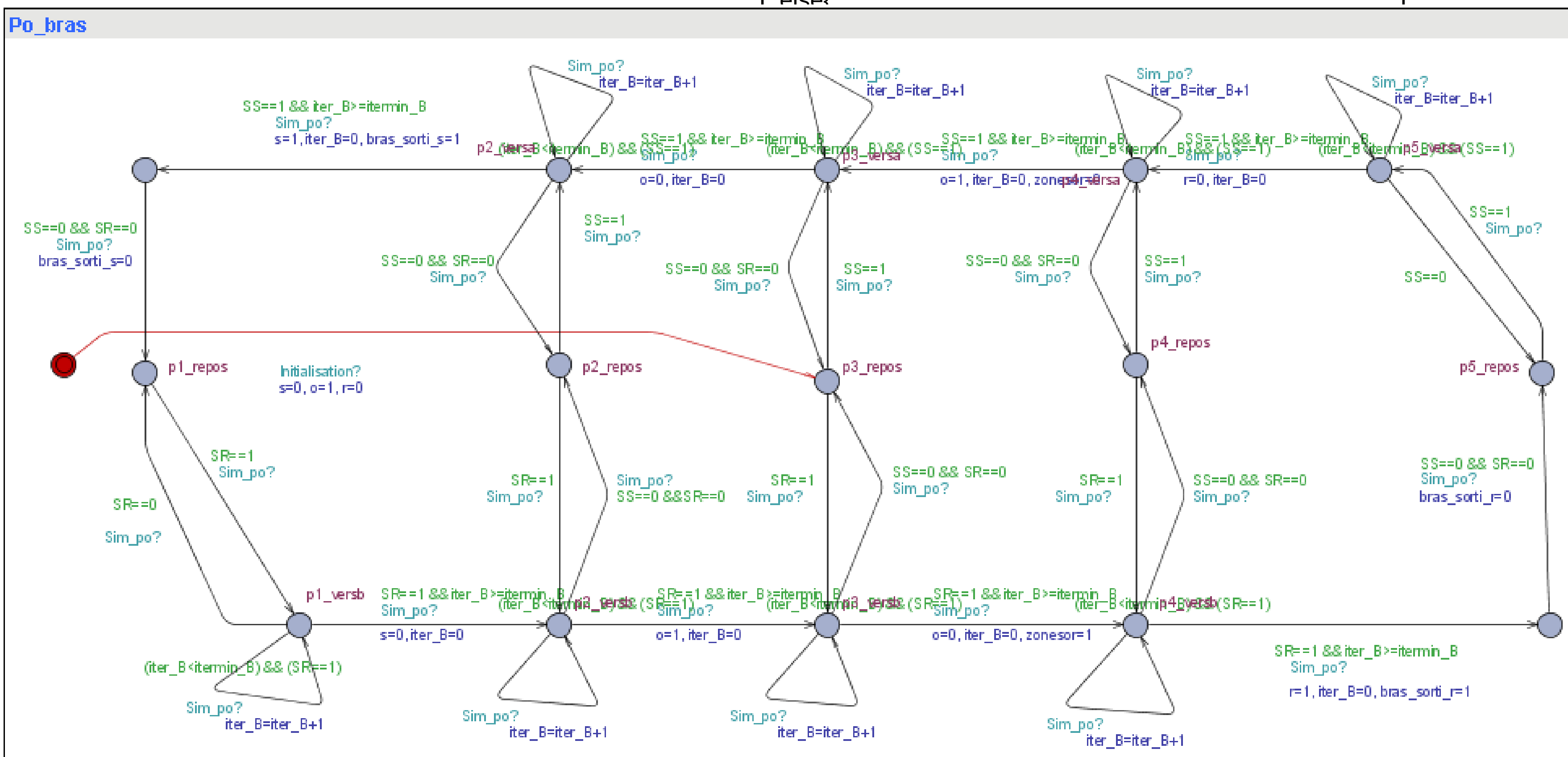
- 3 points to verify :
  - The reachable positions by the transelevator and the forks
  - The possibility of several boxes on a single stock position
  - The possibility of several boxes on the transelevator

Pedagogical applications

Virtual system : Automatic warehouse – Model for the verification

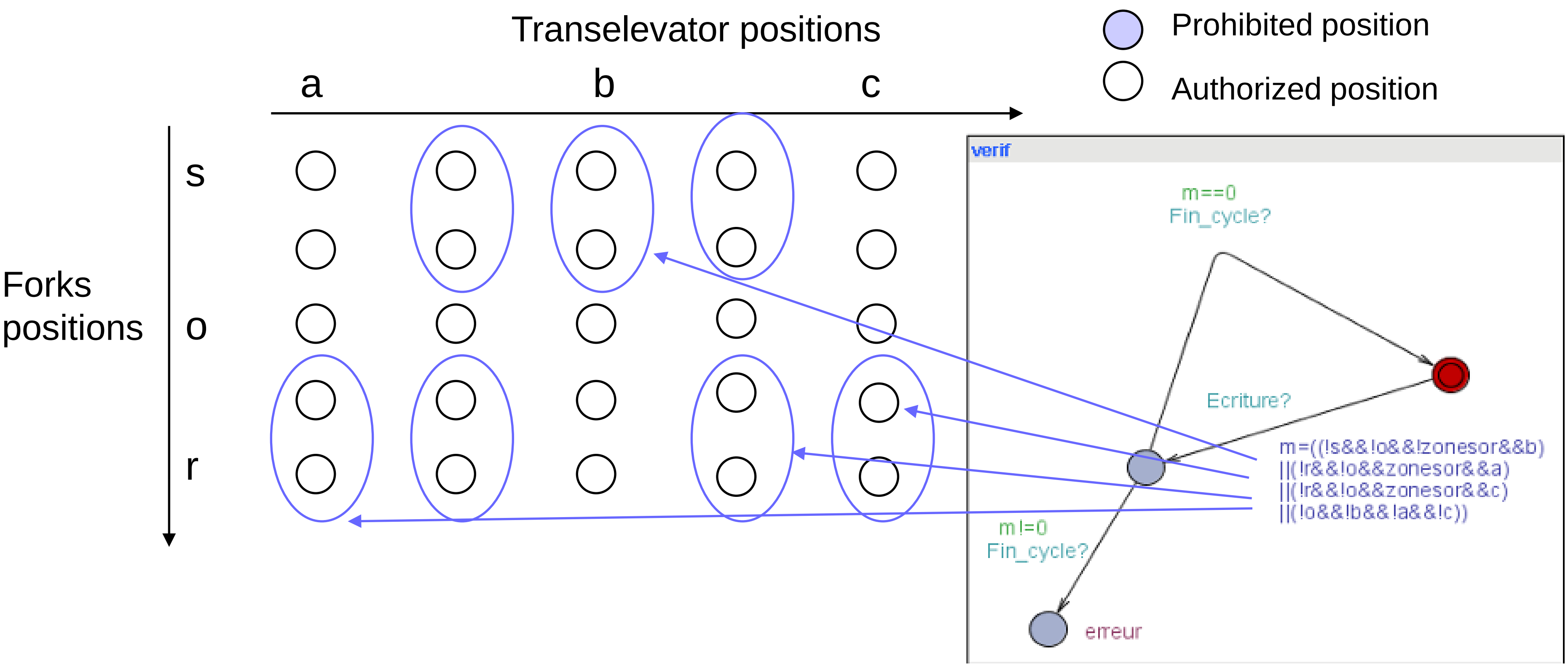
- Model definition
  - For the Forks

```
int vitesse()
{
if((UA&&!UB)==1) {V=V+v;}
else{
```



Pedagogical applications
Virtual system : Automatic warehouse – Model for the verification

- The reachable positions by the transelevator and the forks



- 75 constraints to safe the system
- Test on UPPAAL and implement with PL7_pro

Pedagogical applications
Discussion

- Real system
  - First role of the filter: to safe the system
  - Second role : to bring an explanation in the case of errors
  - To test with various levels of students, to define various levels of system granularity
- Virtual system
  - Role to bring an explanation
  - Show the approach limits, when the sensors number is insufficient

Plan

1. Introduction
2. Control validation approach by filter
3. Constraints verification
4. Pedagogical applications
5. Conclusion and future works

Conclusion and future works
Conclusion

- Questions tackled in this PHD thesis:
  - How to ensure the safety according to the control errors and to check that the control respects the specifications ?
  - How to take into account the human component ?
  - How to ensure that the approach is dependable ?

- Two validation approaches have been proposed
  - Validation approach off line by synthesis
  - Validation approach on line by filter

- To take into account the control designer
  - Functions definition inside the filter

- To guarantee the safety taking care of the implementation
  - Sufficiency verification by model checking,
  - Definition of a system model : computing environment, plant, most permissive control, product

Conclusion and future works
Future works

- Validation approach by filter
  - To develop the functional validation filter
    - To use the Allen's algebra (Allen, 1983)
  - Generation of an explanation adapted to the level of the system validation filter :
    - To adapt the information supplied by the system validation filter
    - To bring an explanation on the safety constraints without redefining the constraints at the functional level
    - To use the traces supplied by UPPAAL for Human adapted explanations

Conclusion and future works
Future works

- Constraints verification approach
  - To verify that all the constraints are necessary
  - To raise the assumptions about inertia and product :
    - Not inertia: Use of temporal information (time of activation of an outputs, outputs sequence …)
    - One product: to consider several products in the same state
  - To propose a hybrid model of the direction model
- To develop a software to define the constraints automatically
- To extend the use of this work within the industrial framework





Synthèse et filtrage robuste de la commande pour des système manufacturiers sûrs de fonctionnement

Synthesis and robust filtering of control for dependable manufacturing systems

Merci pour votre attention

Obrigado para a vossa atenção

Ph.D. for the doctor title of the University of Reims Champagne-Ardenne
By Pascale MARANGE
November 05 2008

Supervised by Bernard RIERA (Professor)
and François GELLOT (Assistant Professor)

CONSEIL REGIONAL CHAMPAGNE ARDENNE
CReSTIC Centre de Recherche en STIC
UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE