PETRI NET AND OO FOR THE MODULAR ANALYSIS OF AN AIRCRAFT LANDING SYSTEM

Villani, Emilia
Dept. Eng. Mecatrônica e de Sistemas Mecânicos, Escola Politécnica, USP
Av. Prof. Mello Moraes, 2201 São Paulo Brasil
evillani@usp.br

Junqueira, Fabrício
Dept. Eng. Mecatrônica e de Sistemas Mecânicos, Escola Politécnica, USP
Av. Prof. Mello Moraes, 2201 São Paulo Brasil
fabri@usp.br

Miyagi, Paulo Eigi
Dept. Eng. Mecatrônica e de Sistemas Mecânicos, Escola Politécnica, USP
Av. Prof. Mello Moraes, 2201 São Paulo Brasil
pemiyagi@usp.br

Valette, Robert
LAAS – CNRS (Laboratoire d'Analyse et d'Architecture des Systèmes)
7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4 FRANCE
robert@laas.fr

Abstract. *This work considers the combined used of Petri nets and object-oriented concepts for the modelling and analysis of control systems. The object modularity is explored in order to deal with the system complexity in both modelling and analysis. According to the proposed approach, the verification of behaviour properties is reduced from a complex proof involving the overall model to a set of simpler proofs involving the model of one or a few objects. Each local proof is made considering a set of hypotheses that should then be proven. Particularly, this paper considers the landing system of the military aircraft Rafale (Dassault Aviation) as a case-study. The aim is to test the proposed approach for real systems with a high level of complexity.*

Keywords:*. Verification, object-orientation, Petri nets, hybrid systems.*

1. Introduction

In the system automation field, the design of control systems has been a constant topic of research. Pushed by the increasing competition among industries, new proposals aim to improve system reliability, reduce costs and time of development, among other things. In this context, the formal verification of behaviour properties of the system plays a fundamental hole. The main problem is how to assure that the control system will present a specified behaviour under the variety of circumstances that it can be submitted while in operation.

In general, there are two ways of verifying behaviour properties for a control system: *theorem proving* and *model checking* [Silva et al, 2001]. In the first case, the aim is to infer or contradict the behaviour property by using logical proofs. In the second case the property is verified by determining all the set of reachable states for the system. The advantage of the first approach is that it is not restricted to finite-state systems. On the other hand, properties which can be automatically proven by available theorem provers are rather restricted because the properties are frequently not decidable. In the second case, the model checking can be automatically executed, however, the set of reachable states grows exponentially with the size of the system, turning unfeasible the analysis of large systems. These problems are more accentuated if instead of considering pure Discrete Event Dynamic Systems, it is also necessary to introduce time variables in the model, and are particularly critical for the case of Hybrid Systems (with both discrete and continuous dynamic). For Hybrid Systems, an additional problem is the non-decidability, i.e., the non-guarantee that, with a finite number of steps the property can be proved using model checking techniques [Alur et al., 1995].

In this context, the authors defined a new analysis approach. The main point of this approach is modularity, i.e., how to decompose the analysis problem in order to handle its complexity (and avoid state explosion problems). For this purpose the object-oriented concepts are used during the modelling phase in order to achieve a modular structure of the modelled system. During the analysis, only the model of one or a few objects is considered at a time. Simultaneously, the approach takes advantage of the user knowledge about the system by allowing the introduction of hypotheses considered by the user as 'reasonable'. These hypotheses reduces the set of reachable states and must be proven in the end by using, for example, theorem-proving methods. As the general case of hybrid systems is the target of the approach, Petri nets are used for representing the discrete aspects and differential equation systems for the continuous ones.

Particularly, the purpose of this paper is to apply this approach for the landing system of the military aircraft Rafale, which has also been used as a case study for testing and comparing other analysis techniques. Due to the scale of the case-study, the models and analysis processes are only partially presented.

This paper is organized as follows. The current problems of tools for automatic verification of behaviour properties are discussed in Section 2, based on results presented by previous works. In Section 3 a brief description of the landing system case-study is introduced. An overview of the modelling approach used by the authors and examples of class modelling are presented in Section 4. Then the analysis approach is presented in Section 5. Section 6 illustrated its application for the verification of a safety property in the landing system. Finally, Section 7 draws some conclusions.

2. Tools for Automatic Verification

The landing system of Rafale (a military airplane made by Dassault Aviation) has been adopted as a case study by the French research group StrQdS (Système Temp-réel Qualité de Service[1]) of the GdR-CNRS ARP (Architecture, Réseaux et systèmes, Parallélisme[2]). The aim of this case-study is to analyse, test and compare techniques, tools and approaches for the verification of behaviour properties.

Some of the results obtained in these studies can be found in [Boniol & Carcenac, 2002]. The following tools were tested: NP-Tools + Lucifer translator [Ljung, 1999], Lesar [Halbwachs, 1992] (available at [Lesar, 2003]), SMV [Clarke et al., 1994] (available at [SMV, 2003]) and UPPAAL [Pettersson & Larsen, 2000] (available at [UPPAAL,2003]). In the first three cases the model is described using the language Lustre, in the last case it is a timed automata. All of them are timed models; the continuous dynamics of the controlled (physical) part is not explicitly taken into account.

Except for the SMV, all the other verification tools presented problems and were not able to verify all the properties considering 3 landing-sets (the landing system is explained in Section 3) and a general context (the behaviour of the pilot and the initial state is unknown). In some cases, better results were obtained considering just one or two landing-sets and a particular context.

These problems are consequence of the state explosion. Although some tools, such as UPPAAL, provide a modular approach for modelling, the analysis is global. The analysis processes enumerate all the possible sequences of events in a global time, resulting in an explosion of the number of scenarios and states. All the tools are indeed based on model checking.

If the continuous dynamic of the controlled part is also considered in the model, then the problems faced by the available tools are even worst. Silva et al [2001] presents a comparison among the tools UPPAAL, HyTech [Henzinger et al, 1997], CheckMate [Silva et al, 2000] and Verdict [Stursberg et al, 1998] using a relative simple example of a batch chemical reactor and the conclusion of this work is that computation complexity restricts their application to fairly small systems.

These problems motivate the development of the approach presented in Section 5.

3. The Rafale Landing-System

The Rafale landing system is composed by 3 landing sets containing each one a door and a landing-gear. A simplified schema of a landing set is presented in Figure 1.

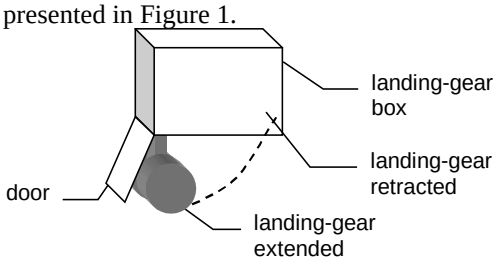


Figure 1. Landing set.

For landing, the following sequence must be performed: open the doors of the landing-gear boxes, extend the landing-gears and close the doors. This sequence is illustrated in Figure 2.

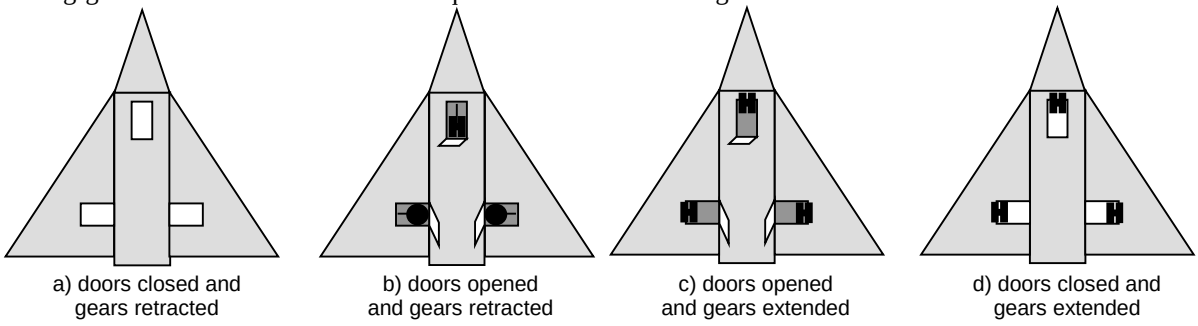


Figure 2. The landing sequence.

[1] http://www.laas.fr/strqds/ (accessed in 24/03/2003)
[2] http://www.arp.cnrs.fr/ (accessed in 24/03/2003)

After taking off, the sequence to be performed is: open the doors, retract the landing-gears and close the doors. An up/down handle is provided for the pilot. When the handle is set UP the extending landing-gear sequence is accomplished, when the handle is set DOWN the retracting landing-gear sequence is accomplished.

The following monitoring components provide information to the pilot:

- A green light informs that the landing-gears are extended and blocked.
- A red light informs that the doors are not closed.
- An alarm panel reports possible faults.
- A state panel informs the current state of the landing-gears and doors.

The landing-gears and doors movement is performed by a set of actuating cylinders:

- For each door, a cylinder (an actuator) opens and closes the door.
- For each right and left landing gear, a cylinder extends and retracts the landing gear, and another cylinder blocks the landing gear in the extended position.
- For the front landing gear, a cylinder retracts, extends and blocks the landing gear in the extended position.

The cylinders are moved by the following electro-valves:

- One general electro-valve that sets pressure on the general hydraulic circuit (the main circuit).
- One electro-valve that sets pressure on the portion of the hydraulic circuit related to door opening.
- One electro-valve that sets pressure on the portion of the hydraulic circuit related to door closing.
- One electro-valve that sets pressure on the portion of the hydraulic circuit related to landing-gear extending.
- One electro-valve that sets pressure on the portion of the hydraulic circuit related to the landing-gear retracting.

Furthermore, in the electric circuit, an analogical relay isolates the computer from the electro-valves. The relay is opened a certain time after the last change made in the up/down handle. This time is supposed to be sufficient for opening or retracting the landing system. When the pilot acts again on the up/down handle, the relay is closed.

On/Off sensors inform the computer about the following positions:

- For each landing-gear, gear and strut actuating cylinder (landing-gear is extended).
- For each landing-gear, landing gear hook (landing-gear is retracted).
- For each door, door catch (door is closed).
- For each door, door actuating cylinder (door is opened).
- General hydraulic circuit (it is pressured).
- Up/down handle (it is UP or DOWN).

Among the tasks of the control software, one is to monitor the state of the system by checking the sensors every 40ms and detecting incoherences on sensor signals (such as closed door sensor ‘on’ and opened door sensor ‘on’). Another task is to execute the UP or DOWN sequence when appropriate, and detect any fault of the system behaviour.

In this case, the overall system (control software + controlled plant) must be analysed in order to ensure that these tasks are accomplished within satisfactory time limits (reachability/liveness properties) and that no undesirable action can be performed (safety properties). Among the reachability/liveness properties, one is to prove that after the last movement of the Up/Down Handle, the landing system will be completely UP or DOWN within 14 seconds. Among the safety properties there is to prove that two valves with opposite action (ex.: open and close door electro-valves) will never be simultaneously opened.

4. The Landing System Modelling

For the verification of the behaviour properties, both control software and controlled system must be considered. The system is hybrid because it presents both events and states of discrete nature and it has a continuous part related to the pressure dynamics on hydraulic circuits and the continuous movement of the actuating cylinders.

The hybrid formalism used for the modelling is the Differential Predicate-Transition Net (DPT net) [Champagnat et al, 1998]. Briefly, in a DPT net:

- A set of variables (x_i) is associated with each token.
- A differential equation system (F_i) is associated with each place (P_i): it defines the dynamic of the x_i associated with the tokens in Pi, according to the time (θ).
- An enabling function (e_i) is associated with each transition (t_i): it triggers the firing of the enabled transitions according to the value of the xi associated with the tokens of the input places.
- A junction function (j_i) is associated with each transition (t_i): it defines the value of x_i associated with the tokens of the output places after the transition firing.

According to the proposed approach the system must be modelled as a set of interacting objects, organized in classes. The modelling approach has already been the focus of other papers [Villani et al, 2002a]. Here only a brief summary is presented. For the introduction of the object-oriented concepts to the DPT net, the following statements are defined, based on class and object concepts of Booch et al [1998]:

- The behaviour of each class is modelled by a DPT net.
- Each object has a set of attributes, which includes its name, the name of the objects it can interacts, and any other variable or parameter used to determine the object dynamic and the interaction with others objects.
- The state of each object is represented by a token (or a set of tokens in a the class net).
- The only way for two objects to interact is by discrete method calls (represented by merging two transitions) or continuous variables sharing (one object can “read” the value of the variables of other objects).

The net of the overall system can be represented in two ways. In a concise representation where there is a single net for each class and all the objects are represented in it. Or this concise representation can be unfolded into a safe net (each place can contain just one token) where each object has its own net and the transition mergings are static. In this paper the second way is used, which is more suitable for the analysis.

The Rafale Landing System is composed by a set of 68 objects organized in 24 classes. From the 24 classes, 7 model the behaviour of the control software. The others are models of the physical components of the landing system, including sensors (1 class), actuators and hydraulic/electrical circuit components (8 classes)[3].

As an example, the model of 4 classes are presented in the following: the *Class 12 - Dedicated Hydraulic Circuit*, the *Class 10 - Positive Pressuring Electro-valve*, the *Class 14 - Electrical Circuit of Positive Pressuring Electro-valve* and the *Class 24 - Dedicated Controller*. The class names are in *italic* and object names are underlined.

In order to understand these models, a more detailed presentation of this part of the system is given.

The actuating cylinders of doors and landing-gears are moved according to the pressure on the hydraulic circuits connected to them, which are called dedicated hydraulic circuit and are modelled as objects of *Class 12*. The two objects of this class in the landing system are *Door Hydraulic Circuit* and *Landing-gear Hydraulic Circuit*. In each dedicated hydraulic circuit, the pressure is controlled by two valves, called positive pressuring valve and negative pressuring valve (*Class 10* and *Class 11*). There are two objects for each of these classes. The Open Door Electro-valve (*Class 10*) and the Close Door Electro-valve (*Class 11*) interact with the Door Hydraulic Circuit (*Class 12*). The Extend Landing-gear Electro-valve (*Class 10*) and the Retract Landing-gear Electro-valve (*Class 11*) interact with the Landing-gear Hydraulic Circuit (*Class 12*).

Figure 3 presents a schema of this part of the landing system. The dedicated hydraulic circuit can assume 3 main configurations: negative pressured, positive pressured and blocked. If the two valves are simultaneously opened the circuit pressure is considered as null and the cylinder behaviour is unpredictable (this situation should never occur in the landing system).

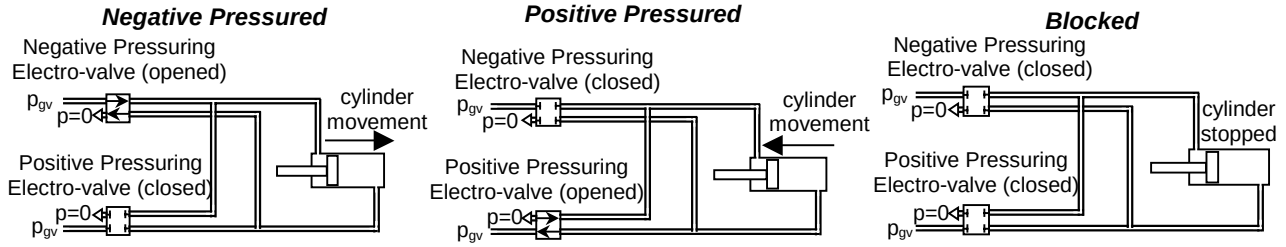
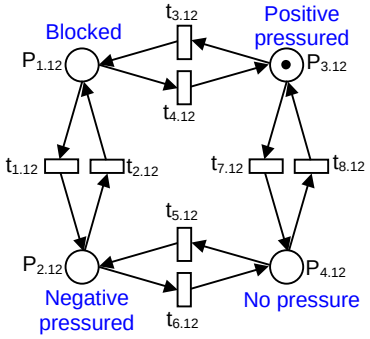


Figure 3. Hydraulic circuit and valves for the actuating cylinders (dedicated hydraulic circuit).

The model of the *Class 12* is presented in Figure 4. The pressure in the hydraulic circuit (continuous variable 'p') is set according to the pressure on a general hydraulic circuit (variable p_{gv_z} from a class not presented here).



External variables :
p_{gv_z}

Junction function
$j_{7.12}$, $j_{6.12}$ – p = 0
$j_{3.12}$, $j_{2.12}$ – p = 0.9*p

Equation system
$f_{1.12}$ – p = p_{gv_z}
$f_{2.12}$ – p = - p_{gv_z}

Methods provided by the class
$t_{1.12}$ – Set negative pressure
$t_{2.12}$ – Block negative Pressure
$t_{3.12}$ – Block positive Pressure
$t_{4.12}$ – Set positive Pressure
$t_{5.12}$ – Eliminate positive pressure
$t_{6.12}$ – Annihilate negative pressure
$t_{7.12}$ – Annihilate positive pressure
$t_{8.12}$ – Eliminate negative pressure

Interaction Parameters
Z – Object of *General Electro-valve/General Hydraulic Circuit*

Figure 4. Model of *Class 12 - Dedicated Hydraulic Circuit*.

The model of the *Class 10* is presented in Figure 5. When the valve is opened, if the dedicate hydraulic circuit is blocked, the valve sets it to positive pressured ($t_{4.10}$ merged with $t_{4.12}$), if the circuit is negatively pressured, the valve sets it to no pressure ($t_{5.10}$ with $t_{6.12}$). When the valve is closed, if the hydraulic circuit is positively pressured, the valve blocks it ($t_{2.10}$ with $t_{3.12}$), if the hydraulic circuit pressure is zero, the valve set it to negative pressured ($t_{3.10}$ with $t_{5.12}$).

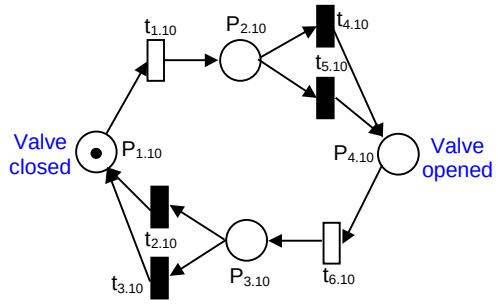
Class 11, which also interacts with the dedicated hydraulic circuit, is similar to the *Class 10*, but instead of setting positive pressure on the dedicated hydraulic circuit, it sets negative pressure (the transition mergings are $t_{2.11/2.12}$, $t_{3.11/8.12}$, $t_{4.11/1.12}$ and $t_{5.11/7.12}$).

Class 14 acts on *Class10*. Its model is presented in Figure 6. The electrical circuit determines if the electrical-valve will be opened or not according to the signal from the control software (*Class 24*) and the position of the analogical relay (not presented here). Basically, the electro-valve is effectively opened (firing of $t_{1.14}$) if the circuit is energized by the control signal (firing of $t_{4.14}$), when it has already been closed by the analogical relay (firing of $t_{7.14}$). If the electro-valve is closed ($P_{2.14}$) when the analogical relay opens the electrical circuit (firing of $t_{3.14}$), it is opened (firing of $t_{6.14}$). In

[3] The complete model is available for downloading at http://www.pmr.poli.usp.br/lsa/.

the landing system there are two objects of this class: the Open Door Electrical Circuit and the Extend Landing-gear Electrical Circuit.

Class 15 is similar to *Class 14*, but instead of acting of *Class 10*, it acts on *Class11*. In the landing system there are also two objects of this class: the Close Door Electrical Circuit and the Retreat Landing-gear Electrical Ciruit.

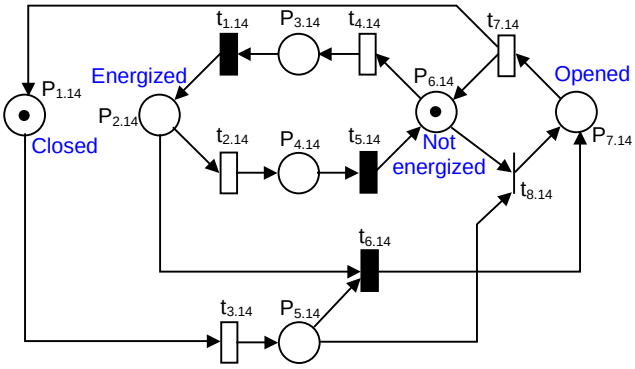


Methods provided by the class
$t_{1.10}$ – Open *Positive Pressuring Electro-valve*
$t_{6.10}$ – Close *Positive Pressuring Electro-valve*

Methods used by the class
$t_{2.10}/t_{3.12.X}$ – Block positive pressure in *Dedicated Hydraulic Circuit*
$t_{3.10}/t_{5.12.X}$ – Eliminate positive pressure *in Dedicated Hydraulic Circuit*
$t_{4.10}/t_{4.12.X}$ – Set positive pressure in *Dedicated Hydraulic Circuit*
$t_{5.10}/t_{6.12.X}$ – Annihilate negative pressure *in Dedicated Hydraulic Circuit*

Interaction Parameters
X – Object of *Dedicated Hydraulic Circuit* class

Figure 5. Model of *Class 10 - Positive Pressuring Electro-valve*.



Methods provided by the class
$t_{2.14}$ – Not energize *Electrical Circuit of Positive Pressuring Electro-valve*
$t_{3.14}$ – Open *Electrical Circuit of Positive Pressuring Electro-valve*
$t_{4.14}$ – Energize *Electrical Circuit of Positive Pressuring Electro-valve*
$t_{7.14}$ – Close *Electrical Circuit of Positive Pressuring Electro-valve*

Methods used by the class
$t_{1.14}/t_{1.10.X}$ – Open *Positive Pressuring Electro-valve*
$t_{5.14}/t_{6.10.X}$ – Close *Positive Pressuring Electro-valve*
$t_{6.14}/t_{6.10.X}$ – Close *Positive Pressuring Electro-valve*

Interaction Parameters
X – Object of *Positive Pressuring Electro-valve*

Figure 6. Model of *Class 14 - Electrical Circuit of Negative Pressuring Electro-valve*.

Class 24 is presented in Figure 7. There are two objects of this class: Door Electro-valve Controller and Landing-gear Electro-valve Controller. Indirectly, they control the objects of *Class 12*.

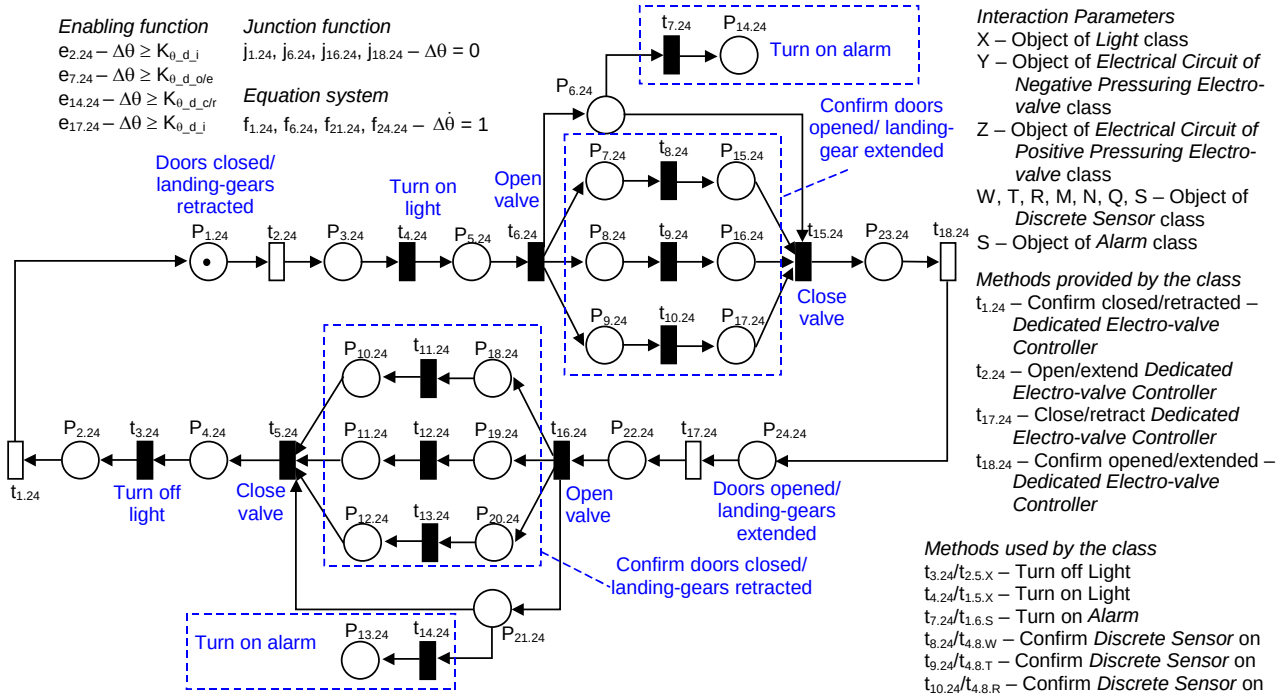


Figure 7. Model of *Class 24 - Dedicated Electro-valve Controller*.

When the method 'Open/extend' is called (firing of $t_{2.24}$), first a light is turned on. Then, the correspondent object of *Class 14* is energized (firing of $t_{6.24}$). The object then waits for the confirmation by sensors that all the doors are opened

or landing-gears extended (firing of $t_{8.24}$, $t_{9.24}$ and $t_{10.24}$). If it not happens in a time interval of $K_{\theta\_d\_o/e}$ then an alarm is turned on. After the confirmation, the controller stops energizing the circuit in order to close the valve and maintain the current position. When the 'Close/retract *Dedicated Electro-valve Controller*' is performed, the same happens but this time the correspondent object of *Class 15* is energized in order to close the doors or retract the landing-gears.

As the scope of this case-study is to analyse the system under nominal operation, the fault detection is included in the models (and a part of the analysis is to prove that no fault is detected under nominal operation), but the diagnosis and treatment are not. It is why nothing is specified when the alarm is turned on. When a fault is detected it is considered that the system enters in a deadlock state.

5. Overwiew of the Proposed Analysis Approach

Two kinds of analysis problems are considered by the proposed approach: the verification of reachability/liveness properties and of safety properties. In the first case, we want to verify that starting from an initial state (completely or partially defined), or from the occurrence of an event, another state will always be reached or another event will always happen. This kind of property can include restrictions upon continuous variables or time durations. In the second case, it is necessary to prove that whatever happens in the system, some forbidden or dangerous states are never reachable.

The most important point of this approach is modularity. Each object, or a small set of object, is analysed at a time. The process starts by considering the objects whose states or transitions are directly concerned by the proof. Taking as an example the verification of safety properties, if the forbidden state is represented by a Petri net place that cannot be marked, then the object containing this place is analysed first. From this forbidden state, a set of scenarios is built by investigating all the possible behaviours that could have led to it. This research is called backward reasoning (because it goes back in time and event sequence). For the reachability properties, the first object analysed is that of the initial state or initial event. From this event or state, the next possible states are determined using the forward reasoning. Once the set of scenarios is defined, a set of conditions about the interactions of this object with others objects (method calls or variable sharing) are determined in order to guarantee the property. Then the objects involved in these interactions must be analysed in order to prove the conditions, and so on.

In both forward and backward reasoning, the equivalence between linear logic and Petri net [Girault et al., 1997] is used to formally represent the discrete part of a scenario as a linear logic sequent. Among the advantages of this association is that it focuses on the causality constraints between the transitions. The next or previous possible events are investigated by using the methods proposed by [Khalfaoui et al, 2001]. This matter is treated with more details in [Villani et al, 2002].

In order to better illustrate the proposed approach, the steps for safety property verification are detailed in the following.

Step 1 – Analysis of Discrete Part – Determining the set of Scenarios by backward reasoning

From the forbidden state, the Petri net is fired backward. Each time that two conflicting transitions are enabled, two different scenarios are generated. The number of transition firings that must be included in each scenario for proving the property is previously unknown. An arbitrary number can be set and if it is not sufficient to prove the property, the overall process of analysis can be repeated including more transition firings.

Step 2 – Analysis of the Continuous Part

Similarly, starting from the possible values for the continuous variables that can satisfy the enabling function of the scenarios' transitions, the set of possible evolutions for these variables is determined considering the equation systems associated with places and the junction function associated with transitions.

Step 3 – Analysis of Object Interaction - Determining the set of conditions for proving the property

For each scenario that can lead to the forbidden state, the interaction of the object with other objects is analysed in order to determine a set of conditions that can invalidate the scenario. These conditions are such that they must turn impossible the firing of one or more transitions of the scenario. Examples of condition are: restrict the value of an external variables in order that enabling function of the transition will never be satisfied, or impose that a method will never be called, etc.

For each scenario, the conditions generate obligations of analysis to other objects. The next steps are performed for each obligation of analysis, and can result in a new set of obligations. They are repeated until no more obligations are left.

Step 4 – Analysis of the Discrete Part - Scenario refinement

For each new object that is analysed, each scenario must be refined in order to include the behaviour of this object. Taking as an example the case when the scenarios built in Step 1 for an Object A contains two transition firings from the interface of Object A with Object B, then the possible behaviours for Object B between this two transition firings must be determined. When more than a possible behaviour is found then the scenario is unfolded into various scenarios.

Step 5 – Analysis of the Continuous *Part*

This step is similar to Step 3. After investigating the possible behaviour of this object for each scenario, if the conditions are not proved, then the analysis process continues by considering the interactions of this object with others.

During the analysis process, hypotheses that are considered as 'reasonable' by the user (the person who is making the analysis) can be temporally assumed as true. They must be proven in the end. The aim of these hypotheses is to

restrict the number of scenarios. They are a based on the user knowledge about the system (interpretation of the model) and cannot be inferred from the model formalism. Each hypothesis works as an intermediate lemma in a mathematical proof: it reduces a complicated proof into a concatenation of elementary proofs.

6. Example of Verification of a Behaviour Property

The property to be proven is that the state "No Pressure" ($P_{4.12}$) of the object Door Hydraulic Circuit of *Class 12* is not reachable. The possible initial states are the landing system up or down. Due to the limited space, some steps of the analysis are not presented[4].

Besides the Door Hydraulic Circuit, this safety proof involves the analysis of the objects Open Door Electro-valve (*Class 10*), Close Door Electro-valve (*Class 10*), Open Door Electrical Circuit (*Class 14*), Close Door Electrical Circuit (*Class 15*) and Door Electro-valve Controller (*Class 24*). The sequence of analysis is a result of the possible interaction among classes and is illustrated in Figure 8.

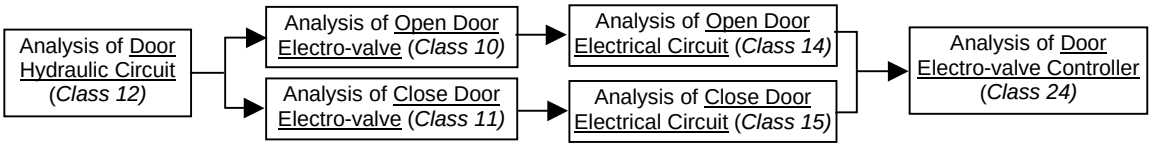


Figure 8. Sequence of analysis.

Analysis of Door Hydraulic Circuit of (*Class 12*)

Step 1 – Analysis of Discrete Part – Determining the set of Scenarios by backward reasoning

Basically, the state "No Pressure" is reached if both the open door and close door electro-valves are concurrently opened. By backward reasoning, two possible scenarios that could lead to this situation are found: first open the Open Door Electro-valve and then open the Close Door Electro-valve (Scenario 1), or exactly the opposite, i.e., open the Close Door Electro-valve and then open the Open Door Electro-valve (Scenario 2). Both are illustrated in Figure 9. In each graph, the nodes are the transition firings and the connections among them represent the conditions for these events (in Scenario 1, transition $t_{4.10/4.12}$ must fire in order to produce the token in $P_{3.12}$, that is a condition for firing $t_{5.11/7.12}$).

Step 2 – Analysis of the Continuous Part

There are no enabling functions associated with the transition firings. It will be assumed that the continuous dynamic of the variable 'p' does not interfere in the proof. If this assumption turns to be false, the analysis of Door Hydraulic Circuit must be done again.

As the local analysis of this object does not prove the property, it means that, if the property is true, it is ensured by the behaviour of the objects that interact with Door Hydraulic Circuit. The next step is to determine what are the conditions that the interacting objects must satisfy in order to reach the forbidden state. At least one of these conditions must be proven as false by analysing these objects.

Step 3 – Analysis of Object Interaction - Determining the set of conditions for proving the property

There are two ways of proving that each scenario will never happen. For Scenario 1, it should be proven that transition $t_{4.10/4.12}$ never fires or that transition $t_{5.11/7.12}$ never fires. For Scenario 2, it should be proven that transition $t_{4.11/1.12}$ never fires or that transition $t_{5.10/6.12}$ never fires. The first possibility for each scenario is probably false because otherwise the hydraulic circuit would never be positively or negatively pressure, which means that the doors would never change its position. The second possibility is more reasonable and therefore is the way chosen to prove this property.

The interface transition $t_{5.11/7.12}$ can fire when it is enabled both in *Class 12* and in *Class 11*. In *Class 12* it is enabled from the firing of $t_{4.10/4.12}$ until the firing of any other transition that consumes the token in $P_{3.12}$. The only possible alternative scenario for Scenarios 1 is Scenario 3. Similarly, the alternative scenario for Scenario 2 is Scenario 4. In Scenario 3, after opening the Open Door Electro-valve, this valve is always closed before the Close Door Electro-valve could be opened. Similarly, in Scenario 4, after opening the Close Door Electro-valve, this valve is always closed before the Open Door Electro-valve could be opened.

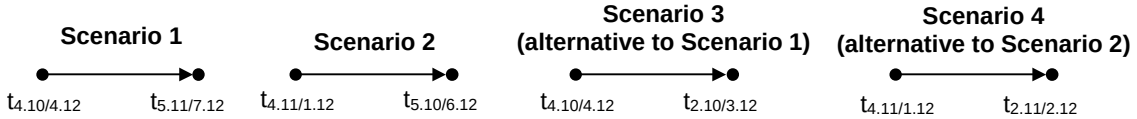


Figure 9. Scenarios for the Door Hydraulic Circuit.

In order to prove that $P_{4.12}$ is never reached, it must be proven that *transitions $t_{2.10/3.12}$ and $t_{2.11/2.12}$ fires before $t_{5.11/7.12}$ and $t_{5.10/6.12}$ get enabled in Class 10 and 11*. In this case, Scenario 3 and 4 will always happen instead of Scenarios 1 and 2. For this, objects of *Class 10* – Open Door Electro-valve and of *Class 11* – Close Door Electro-valve must be analysed.

[4] A complete version can be found in http://www.pmr.poli.usp.br/lsa/, as well as other examples of property verification for the Rafale landing system.

Analysis of the Open Door Electro-valve *(Class 10)*

Step 4 – Analysis of the Discrete Part - Scenario refinement

In Scenario 1, the only transition of the interface *Class 10/Class 12* that is fired, is $t_{4.10/4.12}$. In order to determine when transition $t_{4.10/4.12}$ get enabled in object Open Door Electro-valve, the backward reasoning is applied. The possible scenarios that can lead to the final marking that enable this transition ($P_{2.10}$) are explored (these scenarios cannot contain any other transitions of the interface of *Class 10/Class 12*). In this case the only way of reaching $P_{2.10}$ is by firing $t_{1.14/1.10}$. The same is done for Scenario 2, resulting that the only way of enabling $t_{5.10/6.12}$ is also by firing $t_{1.14/1.10}$.

Scenario 3 contains two transitions of the interface *Class 10/Class 12*: $t_{4.10/4.12}$ and $t_{2.10/3.12}$. The first one is pre-condition for the second. In this case it is necessary to do two local analysis: determine the possible scenarios that can lead to the marking that enable this transition $t_{4.10/4.12}$ ($P_{2.10}$), and determine the possible scenarios that after the firing of $t_{4.10/4.12}$ ($P_{4.10}$) can lead to the marking that enable transition $t_{2.10/3.12}$ ($P_{3.10}$). The results are presented in Figure 10 (Scenario 4 is not presented because it does not contain any transition of *Class 10*).

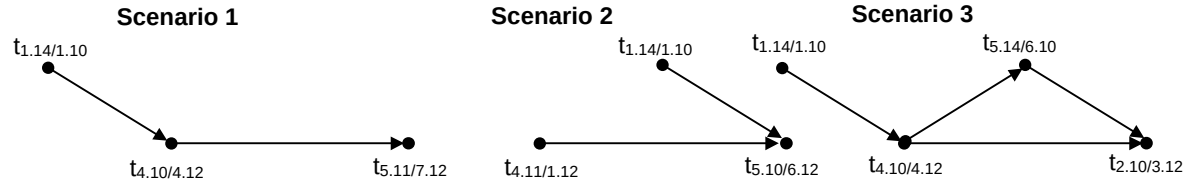


Figure 10. Scenarios 1,2 and 3 including the transition firings of Open Door Electro-valve.

Step 5 – Analysis of the Continuous Part

There is no enabling function associated to the transition firings and no equation system associated to places.

In order to determine the time of the transition firings, the following hypotheses are made (and proven in the end):

Hypothesis 1 – When $t_{1.14/1.10}$ fires, either $t_{4.10/4.12}$ or $t_{5.10/6.12}$ are enabled (and therefore fires instantaneously).

Hypothesis 1 results in $\theta_{4.10/4.12} = \theta_{1.14/1.10}$ (case of $t_{4.10/4.12}$ enabled) and $\theta_{5.10/6.12} = \theta_{1.14/1.10}$ (case of $t_{5.10/6.12}$ enabled)

Hypothesis 2 – When $t_{5.14/6.10}$ fires, either $t_{2.10/3.12}$ or $t_{3.10/5.12}$ are enabled (and therefore fires instantaneously).

Hypothesis 2 results in $\theta_{2.10/3.12} = \theta_{5.14/6.10}$ (case of $t_{2.10/2.12}$ enabled) and $\theta_{3.10/5.12} = \theta_{5.14/6.10}$ (case of $t_{3.10/5.12}$ enabled)

Considering these hypotheses, the proof to be done is rewritten as:

- *For Scenario 1 and 3: After the firing of $t_{1.14/1.10}$ and $t_{4.10/4.12}$, transition $t_{5.14/6.10}$ and $t_{2.10/3.12}$ have to be enabled and fired before transition $t_{5.11/7.12}$.*
- *For Scenario 2 and 4: After the firing of $t_{4.11/1.12}$, transition $t_{2.11./2.12}$ has to be enabled and fired before transitions $t_{5.10/6.12}$ or $t_{1.14/1.10}$.*

The object of *Class 14* – Open Door Electrical Circuitt, must be analysed. Due to the limited space, the next analyses will be summarised.

Analysis of the object of *Class 11, Class 14 and Class 15* – Close Door Electro-valve

The analysis of object of *Class 11* is similar to that of *Class 10*. The final results are:

- For Scenario 1, in order to fire $t_{5.11/7.12}$, $t_{1.15/1.11}$ must be fired.
- For Scenario 2 and 4, in order to fire $t_{4.11/1.12}$, $t_{1.15/1.11}$ must be fired.
- For Scenario 4, in order to fire $t_{2.11/2.12}$, $t_{5.15/6.11}$ must be fired after the firing of $t_{4.11/1.12}$.

By analysing the time of transition firings and making hypotheses, the proof to be done is rewritten as:

- *For Scenario 1 and 3: After the firing of $t_{1.14/1.10}$ and $t_{4.10/4.12}$, transition $t_{5.14/6.10}$ and $t_{2.10/3.12}$ have to be enabled and fired before transition $t_{5.11/7.12}$ or $t_{1.5/1.11}$.*
- *For Scenario 2 and 4: After the firing of $t_{1.15/1.11}$ and $t_{4.11/1.12}$, transitions $t_{5.15/6.11}$ and $t_{2.11./2.12}$ has to be enabled and fired before transitions $t_{5.10/6.12}$ or $t_{1.14/1.10}$.*

Briefly, the analysis of object of *Class 14* and *15* results in a set of conditions for the transition firings. By making a set of hypotheses, it is determined that the Open Door Electro-valve is effectively opened (firing of $t_{4.10/4.12}$) as soon as a command for opening the valve is executed (firing of $t_{6.24/4.14}$), i.e., there is no time interval between the two events. It is also closed (firing of $t_{2.10/2.12}$) as soon as a command for closing the valve is executed (firing of $t_{15.24/2.14}$).

During the analysis of object of *Class 14* the following hypotheses are made:

Hypothesis 3 – When $t_{6.24/4.14}$ fires $t_{1.14/1.10}$ gets enabled in a time interval of $\Delta\theta=0$ (and therefore fires in $\Delta\theta=0$).

Hypothesis 4 results in: $\theta_{1.14/1.10} = \theta_{6.24/4.14}$

Hypothesis 4 – When $t_{15.24/2.14}$ fires $t_{5.14/6.10}$ gets enabled in a time interval of $\Delta\theta=0$ (and therefore fires in $\Delta\theta=0$).

Hypothesis 4 results in: $\theta_{5.14/6.10} = \theta_{15.24/2.14}$

The same is valid for the Close Door Electro-valve. The proof to be done is rewritten as:

- *For Scenario 1 and 3: After the firing of $t_{6.24/4.14,}$ $t_{1.14/1.10}$ and $t_{4.10/4.12}$, transition $t_{15.24/2.14}$, $t_{5.14/6.10}$ and $t_{2.10/3.12}$ have to be enabled and fired before transitions $t_{5.11/7.12}$ or $t_{1.5/1.11}$ or $t_{16.24/4.15}$.*
- *For Scenario 2 and 4: After the firing of $t_{16.24/4.15,}$ $t_{1.15/1.11}$ and $t_{4.11/1.12}$, transitions $t_{5.24/2.15}$, $t_{5.15/6.11}$ and $t_{2.11./2.12}$ has to be enabled and fired before transitions $t_{5.10/6.12}$ or $t_{1.14/1.10}$ or $t_{6.24/4.14}$.*

The last object to be analysed is Door Electro-valve Controller of (*Class 24*), which fires all the transitions involved in the statement).

Analysis of the object of *Class 24* – Door Electro-valve Controller

Step 4 – Analysis of the Discrete Part - Scenario refinement

In the absence of alarm (normal behaviour), there is a unique behaviour of object Door Electro-valve Controller after the firing of $t_{6.24/4.14}$ (Scenario 1 in Figure 11) and a unique behaviour after the firing of $t_{16.24/4.15}$ (Scenario 2 in Figure 11).

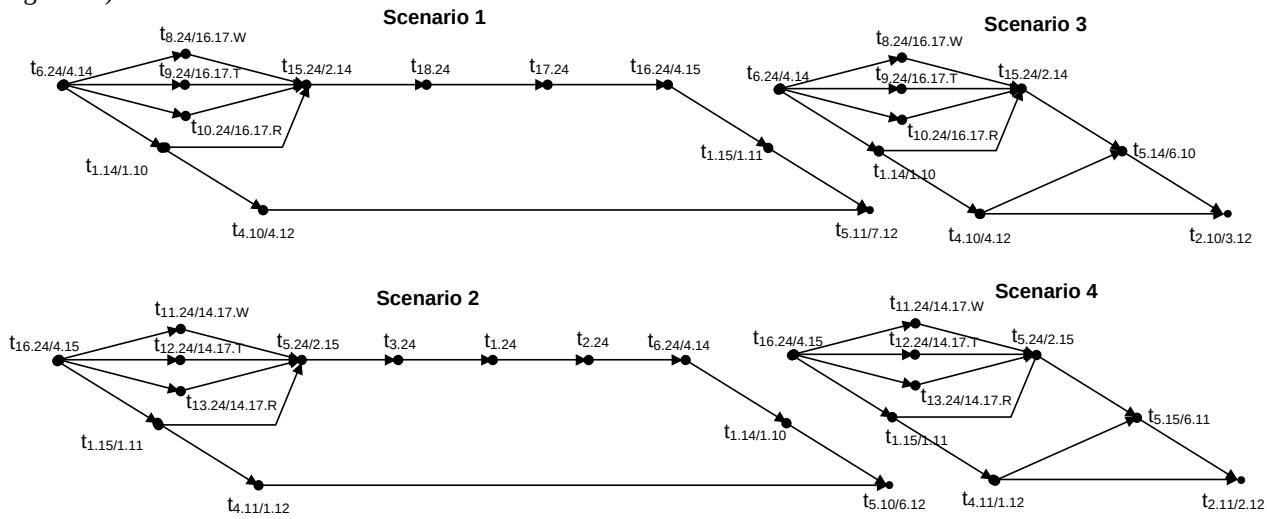


Figure 11. Scenarios 1, 2, 3 and 4 including the transition firings of *Class 24*.

These Scenarios verify the condition that after the firing of $t_{6.24/4.14}$, transition $t_{15.24/2.14}$ is enabled and fired before transition $t_{16.24/4.15}$, and that after the firing of $t_{16.24/4.15}$, transition $t_{5.24/2.15}$ is enabled and fired before transitions $t_{6.24/4.14}$. In order to prove that $t_{5.14/6.10}$ and $t_{2.10/3.12}$ also fires before $t_{15.24/2.15}$, and that $t_{5.15/6.11}$ and $t_{2.11/2.12}$ also fires before $t_{5.24/2.14}$, the continuous part must be analysed.

Step 5 – Analysis of the Continuous Part

By analysing the continuous part, it is verified that there is always a minimum time interval of $K_{\theta\_d\_i}$ (>0) between the command to close the Open Door Electro-valve (firing of $t_{15.24/2.14}$) and that to open the Close Door Electro-valve (firing of $t_{16.24/4.15}$). As the firing of enabled transitions is priority over the time evolution in the DPT nets, transitions $t_{5.14/6.10}$ and $t_{2.10/3.12}$ always fire before transition $t_{16.24/4.15}$ (according to the Hypotheses 1-4, $t_{5.14/6.10}$ and $t_{2.10/3.12}$ get enabled in a time interval of $\Delta\theta=0$ while $t_{16.24/4.14}$ get enabled after $\Delta\theta = K_{\theta\_d\_i} >0$). It means that the Open Door Electro-valve is always effectively closed when the Close Door Electro-valve is opened. The same is valid for the command to close the Close Door Electro valve and that to open the Open Door Electro-valve.

The safety property is therefore proven if the hypotheses that have been assumed are also proven to be true.

Proof of Hypothesis 1, 2

For the proof of these hypotheses, the net of Classes 10, 11 and 12 are fused.

Proof of Hypothesis 1: When $t_{1.14/1.10}$ fires, either $t_{4.10/4.12}$ or $t_{5.10/6.12}$ are enabled (and therefore fires instantaneously).

Hypothesis 1 can be rewritten as '$M(P_{2.10}) \leq M(P_{1.12}) + M(P_{2.12})$'. In the net resulting from the fusion of the net of Classes 10 and 12, the following place invariant is identified: '$M(P_{1.10}) + M(P_{2.10}) - M(P_{1.12}) - M(P_{2.12}) = 0$', according to the initial markings. Therefore '$M(P_{2.10}) = - M(P_{1.10}) + M(P_{1.12}) + M(P_{2.12})$' and Hypothesis 1 is proven.

Proof of Hypothesis 2: When $t_{5.14/6.10}$ fires, either $t_{2.10/3.12}$ or $t_{3.10/5.12}$ are enabled (and therefore fires instantaneously).

Hypothesis 2 can be rewritten as '$M(P_{3.10}) \leq M(P_{3.12}) + M(P_{4.12})$'. In the net of, the following place invariant is identified: '$M(P_{3.10}) + M(P_{4.10}) - M(P_{3.12}) - M(P_{4.12}) = 0$', according to the initial markings. Therefore '$M(P_{3.10}) = - M(P_{4.10}) + M(P_{3.12}) + M(P_{4.12})$' and Hypothesis 2 is proven.

The proof of Hypothesis 3 and 4 are similarly done by fusing the nets of *Classes 14* and *10*. In order to make this proof, the following hypothesis must be made: the Open Door Electrical Circuit is only opened by the Analogical Relay ($t_{5.16/3.14}$) while the valve is not energized, i.e., $\theta_{5.16/3.14} \not\subset [\theta_{6.24/4.14}, \theta_{15.24/2.14}]$, which means that $t_{6.14/6.10}$ never fires. The proof of this last hypothesis is involve objects of classes not presented in Section 4, and therefore is not discussed here.

Although the proof of this behaviour property is relatively simple, it shows how the state explosion problem can be avoided. Taking Scenario 1 as an example, in a global approach it would be necessary to consider all the possible events in the other 62 objects between $t_{6.24/4/14}$ and $t_{5.11/7.12}$. If the scenario duration were of a few seconds, it would include thousands of events related to the coherence checking that are executed by the control system with a frequency of milliseconds. Furthermore, different scenarios would be created for considering the possible orders of independent events, such as the firing of $t_{9.24}$, $t_{10.24}$, and $t_{11.24}$. For the property verification it is not relevant if $t_{9.24}$ fires before or after $t_{10.24}$, it is only important to consider that both of them fires after $t_{6.24}$ and before $t_{15.24}$. The focus on the causality among events instead of on the global sequence of events is possible due to the use of linear-logic as formalism, this issue is detailed in [Villani et al, 2002b].

7. Conclusion

In this paper a new modular approach is introduced for the verification of properties in control systems. The aim of the approach is to avoid large models of the system where the state of all its components are considered and where global sequences of the events must be defined. Instead of it, the proposed approach analyses each object (or small sets of objects) at a time and focuses on the causality constraints between the transitions. In this manner, the state explosion problem can be avoided.

It is important to highlight that the overall analysis approach cannot be automated because it needs the user interaction (and therefore it cannot be entirely performed by a computational tool). However, the purpose of this approach is not to be faster or easier to use than automatic tools, but to provide an alternative solution for the cases where these tools turn out to be unable to solve the problem. Simultaneously, some steps of the approach can still be automated (such as the building of discrete scenarios and the computation of place invariants) and, among the future tasks, the development of a supporting computational tool is considered. Its purpose is to aid and guide the user throughout the analysis process, merging user knowledge of the system and computer processing capability in a synergetic way in order to solve complex problems.

At present, the approach is being applied to a number of examples in order to identify its limits and the kind of problems to which it is better applicable. In addition to the landing system, which presents complex discrete interaction among objects, another case-studied considered is the supervision of a cane sugar factory. Contrarily to the landing system, this case-study has objects with complex continuous dynamic and numerous continuous variable sharings.

8. Acknowledges

The authors would like to thank Dassault Aviation for gently providing the case-study of this paper. This research is partially supported by governmental agencies FAPESP, CNPq and CAPES.

9. References

Alur, R. et al., 1995. “The algorithm analysis of hybrid systems”. *Theorical Computer Science*, vol.138, pp 3-34.

Boniol, F. & Carcenac, F., 2002. “Une étude de cas pour la vérification formelle de propriétés temporelles”. *Journées Formalisation des Activités Concurrentes*, Toulouse.

Booch, G., et al. 1998. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, Inc. Harlow, England.

Champagnat, R., 1998. “Supervision des Systèmes Discontinus: Definition d’un Modèle Hybride et Pilotage en Temps-rèel” Thèse de Doctorat, Université Paul Sabatier, Toulouse, France.

Clarke, E. et al., 1994. “Verification tools for finite-state concurrent systems”. *In: A Decade of concurrency--Reflections and Perspectives . Lecture Notes in Computer Science*, vol. 803.

Girault, F. et al., 1997. “A logic for Petri nets”. *JESA* vol. 31, n. 3, Editions Hermes.

Kalfaoui, S. et al., 2001. “Extraction des scenarios critiques à partir d’un modele RdP à l’aide de la logique lineaire”. *Modélisation des Systèmes Réactifs (MSR 2001)*, Toulouse.

Ljung, M., 1999. *Formal modelling and automatic verification of Lustre programs using np-tools*. Master’s thesis, Royal Institute of Technology, Department of Tele-informatics, Stockholm.

Halbwachs, N. et al, 1992. “Programming and verifying real-time systems by means of the synchronous data-flow programming language Lustre”. *IEEE Transactions on Software Engineering, Special Issue on the Specification and Analysis of Real-Time Systems*, vol. 18, n. 9, pp 785-793.

Henzinger, T. A., et al., 1997. “HyTech: A model checker for hybrid systems”. *Software Tools for Technology Transfer*, vol. 1. pp 110-122.

Lesar, 2003. http://www-verimag.imag.fr/SYNCHRONE/lustre.html. Accessed in 24/04/2003.

Petterson, P. & Larsen, K. G., 2000. “Uppaal2k”. In: Bulletin of the European Association for Theoretical Computer Science, vol. 70, pages 40-44.

Silva, B. I. et al (2001) “An Assessment of the Current Status of Algorithmic Approaches to the Verification of Hybrid Systems”. *40th IEEE Conference on Decision and Control*, Orlando.

Silva, B. I., et al., 2000. “Modeling and Verifying Hybrid dynamic systems using checkmate”. *In: Proc. 4th International Conference on Automatic of Mixed Processes: Hybrid Dynamic Systems*, pp 323-328.

SMV, 2003. http://www-2.cs.cmu.edu/~modelcheck/smv.html. Accessed in 24/04/2003.

Stursberg, O., et al., 1998. “Block-diagram based modelling and analysis of hybrid processes under discrete control”. *Journal Europ. des Syst. Automatises*, vol. 32, n.9/10, pp 1097-1118.

Villani, E. et al., 2002a. “An Object-Oriented Approach for Hybrid System Modelling”. *15th IFAC World Congress on Automatic Control*, Barcelona.

Villani, E. et al., 2002b. “Petri nets and Object-Oriented Approach for the Analysis of Hybrid System”. *XIV C. Brasileiro Automatica*, Natal.

UPPAAL, 2003. http://www.docs.uu.se/docs/rtmv/uppaal/. Accessed in 24/04/2003.