

# DIFFERENTIAL PREDICATE TRANSITION PETRI NETS AND OBJECTS, AN AID FOR PROVING PROPERTIES IN HYBRID SYSTEMS

**E. Villani\*, J. C. Pascal<sup>+</sup>, P. E. Miyagi\*, R. Valette<sup>+</sup>**

*\* Escola Politécnica, University of São Paulo*

*Av. Prof. Mello Moraes, 2231 CEP 05508-900 São Paulo, BRAZIL*

*<sup>+</sup> Laboratoire d'Analyse et d'Architecture des Systèmes – LAAS / CNRS*

*7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4 FRANCE*

*e-mail: [evillani@usp.br](mailto:evillani@usp.br), [jcp@laas.fr](mailto:jcp@laas.fr), [pemiyagi@usp.br](mailto:pemiyagi@usp.br), [robert@laas.fr](mailto:robert@laas.fr)*

**Abstract:** This paper introduces a new approach for the verification of behaviour properties in hybrid systems. By using Petri nets and object oriented concepts the proof of a system property 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 case of proving safety properties. *Copyright © 2002 IFAC*

**Keywords:** Petri nets, object modelling techniques, differential equations.

## 1. INTRODUCTION

The increasing employment of system integration and computer automation in industrial systems has lead to the need of dealing with more and more complex hybrid system (Antsaklis & Koutsoukos, 1998). (Here, the term “hybrid” indicates systems that involve both discrete and continuous dynamic) As result of this trend, both modelling and analysis of such systems cannot be easily addressed by the techniques defined for simple applications.

Within the domain of system analysis, one of the most important aspects is the guarantee of the system reliability by the verification of behavioural properties. An example is proving that a forbidden state will never be reached. However, most of works already published can only be applied to special classes of hybrid systems. For the verification tool UPPAL (Amnell et al, 2000), the model must be reduced to a timed automata. Other approaches are based on linear hybrid automata, such as (Gueguen & Zaytoon, 2001) and the verification tool HyTech (Henzinger et al, 1997). Only a few approaches support non linear models, such as the verification tool Checkmate (Silva et al, 2001), which uses non linear hybrid automata but cannot easily deal with large-scale systems (Silva et al, 2001).

The main problem of hybrid system analysis is the non-decidability issue, i.e., the non-guarantee that, with a finite number of steps the property can be proved. As it has been proven by (Alur et al, 1995), if continuous variables with different growing rates (different derivatives) are included in the model, then the reachability may become undecidable. Generally, this is the case of hybrid systems.

In this context, the aim of this paper is to introduce a new approach for the hybrid system analysis. On the contrary of the cited works, Petri nets are used for modelling of the discrete part, and linear logic is used for its analysis, in order to deal with the discrete state explosion problem. For the continuous part, differential equation systems are adopted. The main innovative point of the proposed approach is that it uses the object-oriented concepts to decompose and analyse the system. By this way, an analysis problem, that would otherwise involve the overall model of the system, is decomposed into a set of simpler analysis problems involving the model of one or a few objects. Another important point of the approach is that it is not entirely automated. A more balanced solution is proposed where the user knowledge of the system is used in order to restrict the solution space and avoid the non-decidability (although no guarantee of a solution can be given).

This paper is organised as follows. Section 2 introduces the object-oriented concepts to the Differential Predicate Transition Petri nets using as an example a sugar production process. In Section 3 the analysis approach is presented and, in Section 4, it is illustrated by the verification of a safety property for the example of Section 2. Finally, Section 5 draws some conclusions.

## 2. THE MODELLING APPROACH

### 2.1 Differential Predicate Transition Petri Nets (DPT Petri Nets) and Object Oriented Concepts.

The modelling approach has already been introduced in (Villani et al, 2002). In this paper just a short overview is presented. Briefly, a DPT Petri net defines an interface between differential equation systems and Petri net elements. Its main features are (Champagnat et al, 1998):

- 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  $P_i$ , according to the time ( $\theta$ ).
- 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  $x_i$  associated with the *tokens* of the input *places*.
- A junction function ( $j_i$ ) is associated with each *transition* ( $t_i$ ): it defines the value  $x_i$  associated with the *tokens* of the output *places* after the *transition* firing.

For the introduction of the object-oriented concepts to the DPT Petri net, the following statements are defined, based on class and object concepts of (Booch et al, 1998):

- The behaviour of a class is modelled by a DPT Petri net.
- The attributes of the class is modelled as the set of variables of the DPT Petri net.
- The first variable of a *token* tuple of variables in a class net is the identity of an object.
- An object is represented by a *token* in the class net, or by a set of *tokens* with the same identity.

The communication among objects can be discrete or continuous. The discrete interactions are represented by method calls (Paludetto, 1991). The continuous interactions are modelled by sharing continuous variables among objects. The value of the shared variables is determined by one object and can be used in the junction function, the equation systems or the enabling function of other objects.

### 2.2 The Cane Sugar Production as an Example

The example used to illustrate the proposed approach is part of a sugar production process (Figure 1). The cane juice arrives at the clarifier. Here, the juice passes through a number of compartments where particles settles. The resulted clear juice is sent to the evaporator to produce syrup. Evaporation is a continuous process where the sucrose concentration

increases. The continuous flow of syrup is then stocked in a tank that acts as a capacity element before the crystallisation in the vacuum pans, which is a batch process. An on/off valve controls the flow from the tank to the vacuum pans.

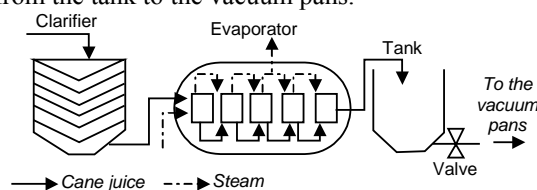


Figure 1. Part of the sugar production process.

A possible model for this system is composed by 4 classes: *Tank*, *On/Off Valve*, *Evaporator*, *Clarifier*.

#### Class Tank model (Figure 2)

The tank volume (variable  $V$ ) can vary between 0 and 100. Its value depends on the incoming flow ( $q_e$  - from the evaporator) and on the outgoing flow ( $q_v$  - to the vacuum pan). When in the state “Normal” ( $P_5$ ), if the volume reaches the upper limit of 80, the tank calls a method of the *Clarifier* class and goes to the “Alert” state ( $P_8$ ). If  $V$  continues to grow up and reaches the threshold of 100, tank goes to the state “Overflow” ( $P_9$ ), which is a dead state (the system cannot return to the state “Normal” without external intervention). On the contrary, if  $V$  goes under the lower limit of 60 then another method of *Clarifier* class is called and the tank returns to the state “Normal”. A similar reasoning is made to avoid the state “Empty” ( $P_1$ ), another dead state. In this case the methods called are of *On/Off Valve* class.

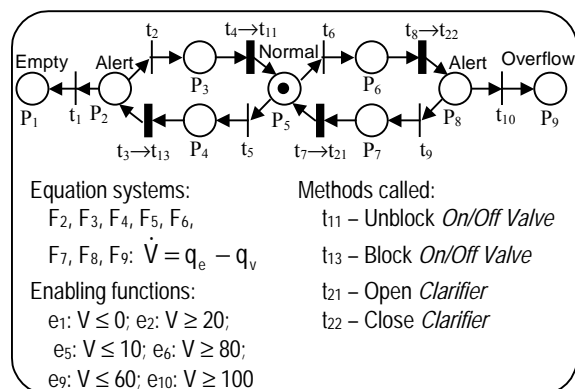


Figure 2. Model of the *Tank* Class.

#### Class ON/OFF Valve model (Figure 3)

The *On/Off Valve* can assume two main states: “Blocked” ( $P_{10}$ ) and “Not Blocked” ( $P_{11}$ ). When the valve is not blocked, it switches between “Opened” ( $P_{13}$ ) and “Closed” ( $P_{14}$ ). The time intervals between the switching are determined by the auxiliary variable  $\theta_{aux}$ . Although a call of the method “Block valve” is immediately accepted (firing of  $t_{13}$ ), the valve is effectively blocked (firing of  $t_{12}$ ) only when the valve goes to the “Closed” mode. This is because the supply of syrup to the vacuum pan could not be interrupted without damage to the production.

#### Class Evaporator model (Figure 4)

The *Evaporator* is modelled by a single differential equation. 30% of the incoming flow ( $q_c$ ) is evaporated. The resulting 70% is sent to the *Tank* with a time constant of 5. Roughly, this time constant

means that a variation on the flow of juice entering in the *Evaporator* will result in a variation on the juice leaving the evaporator ( $q_e$ ) after a certain delay.

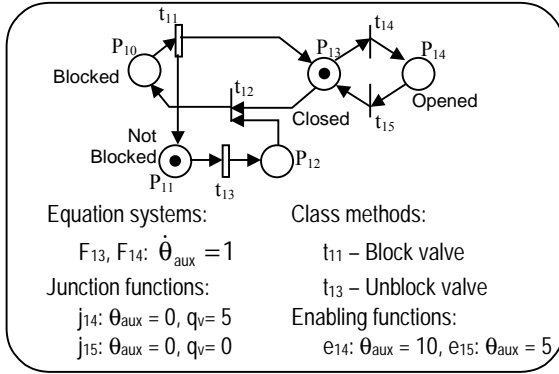


Figure 3. Model of the *On/Off Valve* Class.

$$\text{Equation system: } \dot{q}_e = \frac{0.7 * q_c - q_e}{5}$$

Figure 4. Model of the *Evaporator* Class.

### Class Clarifier

A simplified model for the *Clarifier* is presented in Figure 5. The juice flow leaving the *Clarifier* is proportional the quality factor of the juice ( $Q$ ), which is an external variable.

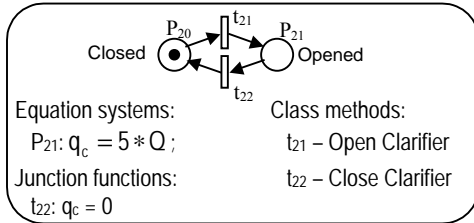


Figure 5. Model of the *Clarifier* Class.

## 3. THE ANALYSIS APPROACH

### 3.1 Decomposition of the Analysis Problem

As presented in the introduction, the main points of the proposed approach are the decomposition of analysis problems based on the decomposition of the model into objects and the decomposition of the model into a discrete part and a continuous one. Instead of analysing the overall model at once, each object (or a small set of objects) is analysed at a time. When it is necessary to reason globally on a set of objects, it will be possible to only consider the discrete part or the continuous one, avoiding to deal with the two ones at the same time. When verifying a property for an object, a set of hypotheses is made about the interaction with other objects. Then the property will be true in the analysed object if the set of hypotheses are also proven to be true. The hypotheses are “proof obligations” therefore the initial proof is broken down into a sequence of local or simpler proofs.

The proof obligations are typically generated by the possible interactions with other objects (method calls or shared variables). The *UML Collaboration Diagram* of the set of objects illustrates the possible sequences of proof obligation for an analysis problem. Considering the example of Section 2.2, the

*Collaboration Diagram* is presented in Figure 6. The method call is modelled by a continuous arrow ( $\longrightarrow$ ), while the variable sharing is modelled by a hatched arrow ( $\dashrightarrow$ ).

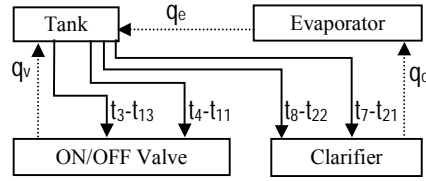


Figure 6. Collaboration Diagram of the example.

Each interaction of the *Collaboration Diagram* (each arrow) can result in a proof obligation. If a property is being verified for object *Tank*, proof obligations can be generated for object *Evaporator* (regarding  $q_e$ ), *ON/OFF Valve* (regarding  $q_v$  and the firing of  $t_{13}$  and of  $t_{11}$ ) and *Clarifier* (regarding the firing of  $t_{22}$  and of  $t_{21}$ ). When verifying the proof obligation for the *Evaporator* another proof obligation can be generated for the *Clarifier* (regarding  $q_c$ ).

### 3.2 Analysis principle

Generally, the verification of behaviour properties for hybrid systems can be classified into two main groups: safety property verification, i.e., to prove that a state or a set of states cannot be reached, and liveness property verification, i.e., to prove that a state, or a set of states, can always be reached. In both cases, for the proposed approach, if the property concerns the state and variables of a single object then this object is the first to be analysed, else the set of objects concerned must be fused and analysed as a single one (under the form of a compound object).

In the analysis of a single object, the linear logic is used as a formalism to explore the possibilities for the object evolution. For the equivalence between Petri nets and linear logic, it is considered the work of (Girault et al., 1997). According to it, a “sequent” in linear logic expresses the reachability from an initial *marking* of the Petri net to a final *marking* by means of firing a set of *transition*. The proof of the sequent is made by using a set of rules allowing verify that the sequent is correctly written (syntactic proof) and is equivalent to the reachability proof for the Petri net. The interesting point is that *transition* firings are not necessarily considered in sequence, a partial order among them is derived from the proof. Regarding the methods for exploring the possible scenarios of a Petri net, it is adopted the approach of (Khalfaoui et al, 2001). Two basic ways for exploring the Petri net evolution are defined: forward reasoning and backward reasoning. In the forward reasoning the initial marking is completely or partially known. *Transitions* are fired (by applying the rules of linear logic) in order to determine the set of reachable states. In the backward reasoning the starting point is the final *marking* and the aim is to explore the possible scenarios that can lead to this final state. The backward reasoning is particularly useful for safety proofs and will be the focus of this paper, while the forward reasoning is treated with more details in (Villani et al, 2002).

### 3.3 Backward reasoning

In this context, the methods proposed by (Khalifaoui et al, 2001) are considered with the modular approach presented in 3.1 for hybrid systems (modelled by DPT Petri nets). The main points of the association between Petri nets and linear logic for the backward reasoning are:

- An atomic proposition “p” is associated with each *place* “P” of the net and it represents the presence of one *token* in this place.
- *Markings* as well as pre and post conditions (Pre(t) and Post(t)) are represented by multiplicative formulas. *Transitions* are represented by implicative formulas (with the linear logic implication  $\multimap$ ) of the form:  
Pre(t)  $\multimap$  Post(t).

With this representation, the proof of the sequent:

$$M_0, \sigma \vdash m_1, \Delta$$

is used to derive a partially ordered list of events  $\sigma$  such that the partial marking  $m_1$  is reached. The partial *marking*  $\Delta$  corresponds to a set of *tokens*, which are necessarily produced when  $m_1$  is produced (side effect). The initial *marking*  $M_0$  is such that the *transition* firings in  $\sigma$  produce all the *tokens* in  $m_1$  and  $\Delta$ . As we deal with hybrid systems, the possible (from the discrete point of view) partially ordered list of events  $\sigma$  has to be consistent with some continuous initial state and continuous dynamics.

### 3.4 Basic steps of the analysis

Following, the steps of the proposed approach are:

*Step 1* – Build a scenario in the object (by means of a backward reasoning). By analysing all the causal relations, this step points out how the forbidden state can be reached from a normal state. During this analysis, all the state changes which are necessary for the occurrence of the scenario (side-effects) are pointed out.

*Step 2* – Establish a list of hypotheses which are likely to help the proof. Each one is then considered as a proof obligation. They have to be proven (as lemmas). They may involve the initial object or objects connected to the initial one either by means of discrete interactions (pointed out at the preceding step) or continuous ones (shared continuous variables) as represented in the *UML Collaboration Diagram*.

*Step 3* – Prove that each scenario leading to the forbidden state is impossible when the continuous dynamics is taken into account. This impossibility may be derived from two cases:

- either by proving that a *transition* which has to be fired is conflicting with another one such that its enabling function will always be true first,
- or by proving that the continuous dynamics associated with one *place* is such that after the firing of a *transition* producing a *token* in this *place*, it turns impossible to fire the *transition* consuming this *token* in the defined scenario.

## 4. ANALYSIS OF THE EXAMPLE

### 4.1 General hypotheses

For the example of 2.2, the safety property to be verified is the non-reachability of state  $P_9$ , (tank overflow). The following assumptions are made:

- The initial volume ( $V_0$ ) is under or equal to 80.
- The initial value of  $q_e$  is within the interval  $[0, 3.5]$ .
- The initial states of the *Tank*, *On/Off Valve* and *Clarifier* are coherent: if  $P_8$  or  $P_9$  is marked in *Tank* then the *Clarifier* is off, else the *Clarifier* is on. If  $P_2$  or  $P_3$  is marked then the *On/Off Valve* is blocked.
- The quality factor  $Q$  can vary between 0.8 and 1.

### 4.2 Step 1 of the method: building the scenario

#### Analysis of Tank Object

The property to be proven is that it is not possible from a normal state (*place*  $P_5$ ) to reach the overflow state (*place*  $P_9$ ). The sequent to be analysed is ( $M_0$  is such that there is a *token* in  $P_5$ , the remaining part is unknown):

$$M_0, \sigma \vdash p_9, \Delta$$

The only enabled *transition* is  $t_{10}$  (for backward firing). Firing of  $t_{10}$  results in  $\sigma = t_{10}, \sigma'$  and a new sequent to be proven:

$$M_0, \sigma' \vdash p_8, \Delta$$

Now, the only backward enabled *transition* is  $t_8$ , but this *transition* corresponds to a synchronization with the object *Clarifier* (*transition*  $t_{22}$  fired). In order to be able to backward fire the pair of *transitions*  $t_8$  and  $t_{22}$ , it is necessary to consider that  $\Delta = p_{21}, \Delta'$ . We have therefore to prove:

$$M_0, \sigma' \vdash p_8, p_{20}, \Delta'$$

After the backward firing of  $t_{8/22}$ , by noting  $\sigma'' = t_{8/22}, t_{10}, \sigma''$ , it is necessary to prove:

$$M_0, \sigma'' \vdash p_6, p_{21}, \Delta'$$

Finally, the backward firing of  $t_6$  leads to:

$$M_0, \sigma''' \vdash p_5, p_{21}, \Delta'$$

with  $\sigma = t_6, t_{8/22}, t_{10}, \sigma'''$ .

It is now possible to prove this sequent if  $M_0$  contains just one *token* in  $P_5$  and one in  $P_{21}$  and if  $\sigma'''$  and  $\Delta'$  are empty. So the scenario leading from the normal state to the forbidden state “overflow” corresponds to the sequent (with  $M_0 = p_5, p_{21}$ ):

$$M_0, t_6, t_{8/22}, t_{10} \vdash p_9, p_{20}$$

It is the unique possible scenario (there is no other way to reach  $P_9$ ).

### 4.3 Step 2 of the method: Listing all the hypotheses (generating proof obligations)

The following hypotheses are made in order to prove the unfeasibility of the scenario:

- H1: *The proposition  $V \leq 80$  is always true when  $M(P_5) = 1$ .*

This hypothesis is reasonable because it is a state

considered as normal. It seems necessary because the *Tank* overflow can be avoided if the *Tank* can absorb the incoming flow ( $q_e$ ) between the firing of  $t_6$  and the effective cease of the flow  $q_e$ . If  $V=100$  when  $t_4$  or  $t_7$  are fired, the overflow *place* seems likely to be the next state. A straightforward consequence of this hypotheses is that  $V = 80$  when  $t_6$  fires.

- H2:  $M(P_8) = 1$  implies  $M(P_{20}) = 1$

As a side effect of the scenario, when a *token* is produced in  $P_8$  (object *Tank*), a *token* is also produced in  $P_{20}$  (object *Clarifier*). It is important to know if it remains or not in this *place* during all the time  $P_8$  contains a *token*, i.e., if the *Clarifier* remains closed while the *Tank* is on the state Alert ( $P_8$ ).

- H3:  $M(P_6) = 1$  implies  $M(P_{21}) = 1$

It is important to know if there will be a delay between the firing of  $t_6$  and that of  $t_{8//22}$ . According to this hypothesis there is no delay, therefore  $V = 80$  when  $t_{8//22}$  fires.

- H4:  $0 \leq q_e \leq 3.5$

It is important to know a bound for  $q_e$  because the derivate of  $V$  is  $q_e - q_v$  (equation system associated with  $P_8$ ). In order to avoid the *Tank* overflow  $V$  must not increase too much.

#### 4.4 Step3 of the method: Taking the continuous dynamics into account

It is necessary to prove that the actual continuous behaviour is such that the scenario cannot occur. The first kind of situation is that of conflicts. There is one case here: when *place*  $P_8$  contains a *token*, *transitions*  $t_{10}$  and  $t_9$  can be fired. The threshold associated with *transition*  $t_9$  is  $V \leq 60$  and that of  $t_{10}$  is  $V \geq 100$ . When  $t_{8//22}$  is fired  $V = 80$  (from hypotheses H1 and H3), so if for example  $V$  is increasing in *place*  $P_8$ , the presence of  $t_9$  does not prevent *transition*  $t_{10}$  to fire. The only way for proving that the scenario is impossible is to prove that the continuous dynamics in *place*  $P_8$  is such that the firing of  $t_{10}$  is impossible. The continuous dynamics of  $V$  in *place*  $P_8$  involves the objects *Evaporator*, *ON/OFF valve* and *Clarifier* because the derivate of  $V$  depends on  $q_e$  and  $q_v$ , and  $q_e$  depends on  $q_c$  (see figure 6).

In order to reach the state of *Tank* overflow by firing of  $t_{10}$ , the enabling function of  $t_{10}$  ( $V \geq 100$ ) should be true:

$$V(\theta_{10}) = V(\theta_8) + \int_{\theta_8}^{\theta_{10}} (q_e - q_v) \cdot d\theta \geq 100$$

From hypotheses H1 and H3, it is stated that  $V = 80$  when the *token* appears in *place*  $P_8$ , therefore:

$$80 + \int_{\theta_8}^{\theta_{10}} (q_e - q_v) \cdot d\theta \geq 100 \Rightarrow \int_{\theta_8}^{\theta_{10}} (q_e - q_v) \cdot d\theta \geq 20$$

This condition must be respected in order to reach the overflow state, i.e., in order that  $V$  can reach the value 100 when  $M(P_8)=1$ . If there is no  $q_e(\theta)$  and  $q_v(\theta)$  that verify this condition then the overflow state will never be reached and the system is safe.

#### Analysis of ON/OFF Valve Object

As  $q_v(\theta)$  is always positive, the previous condition can be rewritten as:

$$\int_{\theta_8}^{\theta_{10}} q_e \cdot d\theta \geq 20$$

The next object to be analysed is the *Evaporator*.

#### Analysis of object Evaporator:

From the equation system of this object, the previous condition can be rewritten as:

$$\int_{\theta_8}^{\theta_{10}} q_e \cdot d\theta = \int_{\theta_8}^{\theta_{10}} [0.7 * q_c - 5 * \dot{q}_e] \cdot d\theta =$$

$$5 * q_e(\theta_8) - 5 * q_e(\theta_{10}) + \int_{\theta_8}^{\theta_{10}} 0.7 * q_c \cdot d\theta \geq 20$$

According to H4,  $0 \leq q_e \leq 3.5$ , resulting in:

$$5 * 3.5 - 5 * 0 + \int_{\theta_8}^{\theta_{10}} 0.7 * q_c \cdot d\theta \geq 20 \Rightarrow \int_{\theta_8}^{\theta_{10}} 0.7 * q_c \cdot d\theta \geq 2.5$$

As  $q_c$  is a variable from object *Clarifier*, this is the next object to be analysed.

#### Analysis of Clarifier Object:

From H2, it is known that  $q_c = 0$  during all the time  $M(P_8) = 1$ . Therefore:

$$\int_{\theta_8}^{\theta_{10}} 0.7 * q_c \cdot d\theta = 0 < 2.5$$

As the condition is not respected,  $V$  cannot reach the value 100 when  $M(P_8) = 1$  and the scenario is impossible with the defined hypothesis. Now, the hypotheses must be proven.

#### 4.5 Proofs of the hypotheses of step 2

##### H2 and H3 - Analysis of Tank + Clarifier Objects

The proofs of H2 and H3 are proofs involving more than one object, but they uniquely involve the discrete view of the model. The Petri nets of the objects *Tank* and *Clarifier* are fused by merging the *transitions*  $t_8$  and  $t_{22}$  and  $t_7$  and  $t_{21}$  respectively. These hypotheses are a straightforward consequence of the two following p-invariants:

$$M(P_8) + M(P_7) = M(P_{20})$$

$$M(P_1) + M(P_2) + M(P_3) + M(P_4) + M(P_5) + M(P_6) = M(P_{21})$$

##### H4 - Analysis of Evaporator Object

The proof of H4 is only based on considerations over the continuous dynamics. According to object *Evaporator*, the dynamic of  $q_e$  is determined by the following differential equation:

$$\dot{q}_e = \frac{0.7 * q_c - q_e}{5}$$

When  $q_e > 0.7 * q_c$ ,  $\dot{q}_e < 0$ , therefore  $q_e$  decreases, approaching  $0.7 * q_c$ . Similarly, when  $q_e < 0.7 * q_c$ ,  $\dot{q}_e > 0$  and  $q_e$  increases. The denominator of the differential equation (number '5') is the time constant associated with the delay between  $0.7 * q_c$  and  $q_e$ . As far as  $q_e$  approaches  $0.7 * q_c$ , the value of  $\dot{q}_e$  approaches 0, therefore  $q_e$  will never surpasses  $0.7 * q_c$ . If the value of  $q_c$  changes,  $q_e$  instantaneously begins to follow the new value of  $0.7 * q_c$ . As a consequence,  $q_e$  is limited

by the values reached by  $0.7 \cdot q_c$ , i.e.,  $\max(q_e) \leq \max(0.7 \cdot q_c)$  and  $\min(q_e) \geq \min(0.7 \cdot q_c)$  (as long as  $q_e(0)$  is also within this interval). This statement is true independently of the implementation of the object *Clarifier*. If the object *Clarifier* is changed the analysis of object *Evaporator* do not need to be remade. This is one of the advantages of the proposed approach.

#### H4 - Analysis of Clarifier Object

By considering the object *Clarifier*, the highest value of  $q_c$  is reached for  $Q=1$  when the *Clarifier* is 'On' and is  $\max(q_c)=5$ . The lowest value is reached when the *Clarifier* is 'Off' and is  $\min(q_c)=0$ . So the upper and lower bound of  $0.7 \cdot q_c$  are 3.5 and 0 (assumed to be true at the initial state).

#### H1 - Analysis of Tank Object

The proof of H1 is more complex. It indeed requires proving that it is impossible to reach the *place*  $P_5$  in the object *Tank* with  $V > 80$ . The approach is similar to the preceding one, that is it is first necessary to build all the scenarios leading to this place and then to prove that they are inconsistent with the proposition  $V > 80$  when the continuous dynamic is taken into account. It is assumed that the initial value of  $V$  is  $V_0 \leq 80$ . There are two scenarios leading to *place*  $P_5$  (their constructions are not detailed here):

$$M_0, t_9, t_{7/21} \vdash p_5, p_{21}$$

with  $M_0$  consisting in one *token* in  $P_8$  and one *token* in  $P_{20}$ . As  $V \leq 60$  when  $t_9$  is fired and as there is no delay between the firing of  $t_9$  and that of  $t_{7/21}$ , then  $V > 80$  is inconsistent when the *Tank* reaches the normal state by the firing of  $t_{7/21}$ .

The second scenario is:

$$M_0, t_5, t_{3/13}, t_{12}, t_2, t_{4/11} \vdash p_5, p_{13}$$

with  $M_0$  consisting in one *token* in  $P_5$ , one in  $P_{13}$  and one in  $P_{11}$ . When  $t_5$  is fired ( $V \leq 10$ ),  $t_{3/13}$  is fired without delay (the proof is similar to that of H2 and H3). As it is the unique way of reaching  $P_2$ , this means that when  $M(P_2) = 1$  implies  $0 \leq V \leq 20$ . When  $t_2$  is fired we have then  $V = 20$ . *Transition*  $t_{4/11}$  is fired without delay (same proof as above) and  $V > 80$  is inconsistent when the *Tank* reaches the normal state by the firing of  $t_{4/11}$ .

## 5. CONCLUSION

In this paper a new approach is introduced for hybrid system analysis based on the use of Petri nets and object-oriented concepts. By exploiting the object independence principle, a global analysis problem is decomposed into a set of local object proofs. Particularly, this paper considers the proof of safety properties.

This work is still under development. The approach is being applied to a number of case studies in order to identify its limits and the kind of problems to which it is better applicable. The purpose is to develop a systematic method and a set of rules that guides its application. It is important to highlight that the overall analysis approach cannot be automated and, therefore, it cannot be entirely performed by a computational tool. This restriction is not considered as an important limitation because the decidability

problem cannot be solved, which means that any automated approach will also be limited. Instead of that, as an attempt to avoid the non-decidability issue, the approach incorporates the user knowledge of the system through the definition of hypotheses and the use of Petri net properties (such as place invariants or the detection of implicit places – as for H2 and H3). Simultaneously, some steps of the approach can still be automated (such as the building of discrete scenarios by linear logic). As a result, user and computer aid can be incorporated in a synergetic way in order to solve complex problems.

## ACKNOWLEDGES

The authors would like to thank the partial financial support of the governmental agencies FAPESP, CNPq, CAPES and FDTE.

## REFERENCES

- Alur, R. et al. (1995) "The algorithm analysis of hybrid systems", *Theoretical Computer Science*, vol.138, pp 3-34.
- Amnell, T. et al (2000) "UPPAAL – Now, Next and Future" *Proc. of MOVEP'2k: MODelling and Verification of Parallel Processes*, Nantes.
- Antsaklis, P., Koutsoukos, X. (1998) "On Hybrid Control of Complex System: a survey", *3rd International Conference on Automation of Mixed Processes*, Reims.
- Booch, G., et al, (1998). *The Unified Modeling Language User Guide*, Addison-Wesley Longman, Inc. Harlow, England.
- Champagnat, R. et al. (1998) "Modelling and Simulation of a Hybrid System through Pr/Tr PN-DAE Model", *3rd International Conference on Automation of Mixed Processes*, Reims.
- Girault, F. et al. (1997) "A logic for Petri nets", *JESA* vol. 31, n. 3, Editions Hermes.
- Gueguen, H., Zaytoon, J. (2001) "Principes de la vérification des systèmes hybrides", *Modélisation des Systèmes Réactifs*, Toulouse.
- Henzinger, T. A. et al (1997) "HyTech: a model checker for hybrid systems", *9th International Conf. on Computer-Aided Verification*, Haifa.
- 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.
- Paludetto, M., 1991. *Sur la commande des procédés industriels: une méthodologie basée objects et réseaux de Petri*, Thèse de Doctorat, Université Paul Sabatier, Toulouse.
- Silva, B. I. et al (2001) "An Assessment of the Current Status of Algorithmic Approaches to the Verification of Hybrid Systems", *40th IEEE Conf. on Decision and Control*, Orlando.
- Villani, E. et al. (2002) "An Object-Oriented Approach for Hybrid System Modelling", *15th IFAC World Congress on Automatic Control*, Barcelona.
- Villani, E. et al. (2002) "Petri nets and Object-Oriented Approach for the Analysis of Hybrid System", *XIV C. Brasileiro Automatica*, Natal.