17

Fault diagnosis for complex systems using Coloured Petri Nets

L. Rodríguez, E. García, F. Morant, A. Correcher and E. Quiles
Universidad Politécnica de Valencia
Spain

1. Introduction

At present the research areas deeply studied because of security, reliability, viability and economy issues is the fault diagnosis area, which warrants some behavior states of a system, a machine or a process for detection, isolation and recovery faults, or even prevents these.
Currently one of the processes extensively studied in the area of fault diagnosis are the plants and chemical processes as well as systems for the study of renewable energy generation where both control and diagnostic systems for large number of variables become too complex and need new and robust techniques for the diagnosis of complex systems. These methods among others are the techniques for modeling discrete event (SED's) with Petri Nets (RDP's).
The study of these techniques has increased significantly over time due to the large number of applications that have been found. The formalism contributed by the PN's in concepts such as concurrency, mutual exclusion and resource sharing are of special interest and have provided greater capacity and power of representation in the resulting models as those carried out by Finite State Machines (FSM's), (Sampath et al., 1996).
Additionally, the PN's provide the ability of apply techniques of merging places that allow reducing size of the resulting models. This synthesis capacity in the resulting models is more accented with the Coloured Petri Nets (CPN) (Jensen, 1992), designing these ones with the general purpose of being a graphical structure based on PN's useful for specify, design and analyze concurrent systems that contribute to the possibility of applying merging techniques for representation of different concurrent subprocesses that coexist in the same PN graphical structure. The CPN's allow assigning functions in their arcs with lineal transformations capacity, which allows great functional variability to the final model. All these characteristics of synchronism and concurrency of the PN's, additionally to the merging techniques of the CPN's will give the necessary robustness to be applied to fault diagnosis in any complex system.
In this chapter it will be shown the functionality and advantage of the use of Latent Nestling Method of faults using Coloured Petri Nets (García et al., 2008), (Rodríguez et al., 2008) for the isolate and diagnosis faults in complex systems comparing it against other diagnostic techniques. Also, their mathematical formalization in discrete, continuous and hybrid systems and some application example. In the same way it will be presented the possible future research of this diagnostic tool in subjects as intermittent fault diagnosis (Correcher et

al., 2004), condition monitoring (Caselitz et al., 1996) and techniques of structuring as: folding and clustering (Keller, 2000).

2. Latent Nestling Method (LNM)

LNM was proposed according to the procedure used by (García et al., 2008) with the purposed of nestling faults into every place of the initial PN using a folding technique. These Petri Nets for the diagnostic methodology were called, Coloured Petri Net for Fault Diagnosis (DCPNs).
This methodology is defined as:

$$D = (P, T, Pre, Post, M_0, C, PLNf, Tf, PV f) \quad (1)$$

where,

- P is a finite set of places.
- T is a finite set of transitions.
- Pre and $Post$ are input and output arc functions, with an additional argument C_k that is the color of the transition firing T_j, thus:
  $Pre(P_i,T_j/C_k)$,
  $Post(P_i,T_j/C_k)$
  Nevertheless these functions are divided in two subsets, it depending of the transition type, behaviour normal transition or fault transition.
  $TF= Tf \cup Tr$
  where Tf and Tr are the fault and recovery transitions corresponding. As well
  $Pre= Pre^{T} \cup Pre^{TF}$
  $Post=Post^{T} \cup Post^{TF}$
  where the arc function are the following:
  $Pre^{T}: PLNf \times T \rightarrow \mathbb{N}$,
  $Post^{T}: PLNf \times T \rightarrow \mathbb{N}$,
  $Pre^{TF}: PLNf \times Tf \cup PVf \times Tr \rightarrow \mathbb{N}$,
  $Post^{TF}: PVf \times Tf \cup PLNf \times Tr \rightarrow \mathbb{N}$
- M_0 is the initial marking.
- C is the colour set assigned to different identifiers. $C=N\cup f$. N is the subset of coloured tokens representing the normal system behaviour.
- $f = \{f_1, f_2, \ldots, f_i\}$ is the subset of coloured tokens representing fault set.
- $PLNf \subseteq P$ is the subset of fault latent nestling places.
- $PVf \subset P$ is the subset of fault verification places.
- $TF \subset T$ is the transition subset including coloured functions.

Definition 1 A normal transition in a DCPN is **enabled** if each place $PLNf_k$ in ^0T_j meets the condition:

$$m(PLNf_k) \geq Pre(PLNf_k,T_j) \quad (2)$$

The main idea to start is design of PNs for the correct dynamical behaviour of the system, applying the same modeling techniques that are used for generalized PNs. However for complex systems the synthesis capabilities of CPNs can be used in these first modeling steps. The next procedure is use the knowledge expert for nestling the respective faults in each place latent nestling fault $PLNf_i$. These faults can be of simple or multiple nature, fault type f_i or f_if_k. In Fig. 1. is shown the place of latent nestling fault $PLNf_i$, where the notation means the fault type i, j and p that belong to the subnets q and n, with the next marking: $M(PLNf_k) = <\bullet n> + <\bullet q> + <f_i^q> + <f_i^n> + <f_j^n> + <f_p^n>$

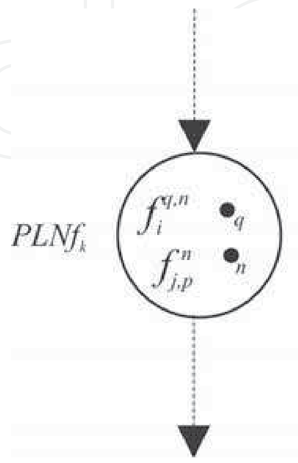

Fig. 1. Latent nestling place

After of the nestling process is necessary realized the trajectories of fault verification as well as fault recovery trajectories. This process is done by building a table, where for each reached marked of the system, exist a fault verification trajectorie by sensors installed.

Definition 2 A fault or recovery transition in a DCPN is **enabled** if each place $PLNf_k$ in ${}^0TF_j$ meets the condition:

for *Tf*:

$$m(PLNf_k) \geq Pre(PLNf_k, Tf_j) \quad (3)$$

for *Tr*:

$$m(PVf) \geq Pre(PVf_k, Tr_j) \quad (4)$$

In Fig. 2. Is shown these trajectories, where if exist a faulty token in the nestling latent place $PLNf_k$, and verify the activation of a non expected reading *SROVnev* drives to the marking of the fault verification place $M(PVf(<f_i^q>))$ by a token of colour f_i^q that evidences the occurrence of this fault type. Consequently, if the fault f_i^q is found in the *PVf*, is possible recovery path to initial *PLNfk* while the expected values *SROVev* are verified. The same way for the f_i^n fault type.

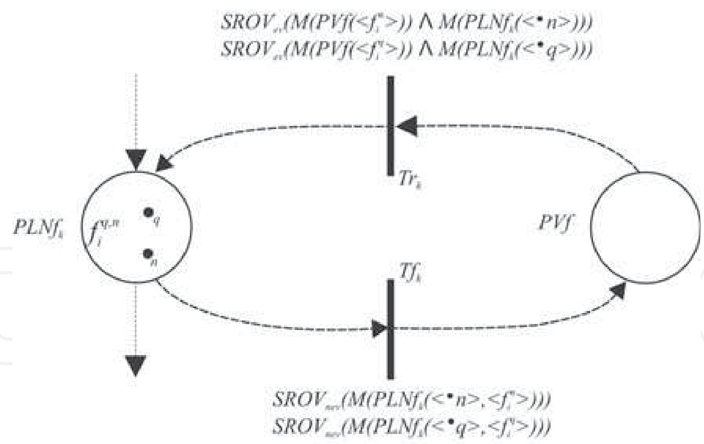

Fig. 2. Verification and recovery faults

The validation of these trajectories would be expressed as:
Verification of a f_i^q fault type:

$$M(PLNf_k(<\bullet q>,<f_i^q>)) \quad [Tf_k=SROVnev(M(PLNf_k(<\bullet q>,<f_i^q>))) \quad >M(PVf(f_i^q)) \tag{5}$$

And f_i^q fault recovery:

$$M(PVf(f_i^q)) \quad [Tf_k=SROVnev(M'(PLNf_k(<\bullet q>)) \wedge M(PVf(f_i^q))) \quad >M(PLNf_k(<\bullet q>,<f_i^q>)) \tag{6}$$

The diagnosability of this methodology is given by following expression:

$$\forall f_i^q \in f \; \exists \; (M(PLNf_k(<\bullet q>,<f_i^q>))) \quad [Tf_k=SROVnev(M(PLNf_k(<\bullet q>,<f_i^q>))) \quad > M(PVf(<f_i^q>)) \tag{7}$$

One question that needs to be asked, however, is that the processes and control systems nowadays are identified by their hybrid and complex nature, allowing the modeling of many methods in different areas of knowledge for process control and fault diagnosis. In this case numerous studies have worked to explain hybrid process in fault diagnosis using differents methodologies, for example (Gertler et al., 1998), (Chen & Patton, 1999), (Patton et al., 1999). As well, authors such as (David & Alla, 2005), analized fault models in Hybrid Petri Nets, other approximation use differential places for represented continuous places with negative markings in a methodology called Differential Petri Nets (Demongodin & Koussoulas, 1998).
For the methology proposal of Latent Nestling, is necessary include places of continuous or differential character, allowing the analysis of continuous dynamical variable. Following, the formal definition and the approaches of this new methodology integrating the use of continuous places. Likewise, an example that clarifies the concepts seen.

3. Latent Nestling Method in Hybrid Systems

3.1 Normal definition

A Hybrid Coloured Petri Net for fault diagnosis (*DHCPN*) is defined as:

$$DH = (P; T; Pre; Post; M_0; Co; C; PLNf; TF; PVf; OS; tempo) \tag{8}$$

Where P; T; M_0; TF; PVf; have the same definition as *DCPN*.
$Pre = PreT \cup PreTF$;
$Post = PostT \cup PostTF$
then,

$PreT : (P \times T) \rightarrow \mathbb{Q}$ ó $\mathbb{N}$,

$PostT : (P \times T) \rightarrow \mathbb{Q}$ ó $\mathbb{N}$,
$PreTF : (P \times Tf \cup PVf \times Tr) \rightarrow \mathbb{Q}$ ó $\mathbb{N}$,
$PostTF : (P \times Tr \cup PVf \times Tf) \rightarrow \mathbb{Q}$ ó $\mathbb{N}$
For every places set can be defined that:

$$P = P^D \cup P^C \tag{9}$$

As well as, PLNf $\subseteq$ P y PVf $\subset P^D$.

Therefore $\mathbb{N}$ corresponds to the case for all $PLNf_i \in P^D$, and $\mathbb{Q}$ corresponds to the case where $PLNf_i \in P^C$. C, remains the coloured tokens set divided into normal behaviours marks "*N*" and representing faults marks "*f*". Being at the same time the normal behaviour marks can be of the discrete or continuous subset, as follows:

$$N = N^D \cup N^C \tag{10}$$

C_0: $P \cup T \rightarrow \{D,C\}$, is a composite function, that indicate for every place of the net if is a latent nestling place of discrete type (set P^D y T^D) or continuous type (set P^C y T^C).
OS: Is a set of operating states, and fault signatures.
This set is defined in the paragraph: trajectories of fault verification and fault recovery.
Tempo: is a delay function that associates a rational number to each transition that can evolve in time, where:

- if $f(T_j)=D$, $tempo(T_j)=d_i=$ is a delay associate at the transition T_j, expressed in time units. As in the method defined in the previous chapter.
- if $f(T_j)=C$, $tempo(T_j)=\{V(T_j), d_i\}=\{V_j, h\}$, V_j represent the maximum firing speed associated with the transition T_j (David & Alla, 2005), and h the firing frecuency represent the sampling time.

This delay function "*tempo*", is implemented for continuous places according to the model characteristics. If the markings and weights of the arcs are not of negative values, only be use the function V_j, represented the maximum maximum firing speed as a constant value according to the degree of *D*-enable. For this case, the function is implemented with a single discrete place associated to the continuous transition that represent the maximum firing speed. In the opposite, if the model has markings or negative arcs, be use the function $\{V_j, h\}$

with a discrete transition associated to a discrete place that is linked to the continuous transition that represents the maximum firing speed. In the last case the behavior of these places and continuous transitions are represented as (Demongodin & Koussoulas, 1998) with the names of differentials places and transitions.

Definition 1. A normal discrete transition in a DHCPN is enable if each place $PLNf_k \in P^D$ in ${}^0T_j^D$ meets the condition:

$$m(PLNf_k) \geq Pre(PLNf_k, T_j^D) \quad (11)$$

Definition 2. A normal continuous transition in a DHCPN is enable if each place $P_i \in P^C$ in ${}^0T_j^C$ meets the condition:

$$m(PLNf_k) \geq Pre(PLNf_k, T_j^C), \quad \text{if} \quad PLNf_k \in P^D \quad (12)$$

or

$$m(PLNf_k) > 0, \quad \text{if} \quad PLNf_k \in P^C \quad (13)$$

or

$$m(PLNf_k) \in \mathbb{Q} \quad \text{if} \quad PLNf_k \in P^C \quad \text{of differential type} \quad (14)$$

3.2 Initial Model

The initial model is the same as in the classic method, unlike that represent differential or continuous places where be model the continuous behavior of the system variables. The first step is to model the behavior of the process, both as discrete and continuous variables involved in the process, it uses the techniques of modeling temporary hybrid systems (David & Alla, 2005). Usually, the discrete processes represent the orders or actions to control the system, while the technological processes are continuous, discrete or mixed. As a second step must be a process of folding into subsystems according to the concurrent of these, this is the the coloured net process, that permit implement normal type marks by each subsystem concurrent global model. This folding process is done using the CPNs techniques. If the model allows it can be implemented directly in CPNs.

3.3 Fault Set Definition

This is done as in conventional nestling, identifying all the faults sets to diagnose in the system and making an allocation of these faults respect to some colored markings of the type $f=\{f_1,f_2,\ldots,f_i\}$.

Furthermore, this set should define continuous type faults to be determined according to the behavior of the residue and thresholds assigned by either the process or expertise.

If a fault f_i occurs from an abnormal behavior of a continuously variable h, being the continuous place is influenced by a normal behavior mark q contain in a $PLNf_k$. The fault is designated as a pair $< f_i^q, S_i >$, where f_i is the fault occurred in the subnet q, and S_i is the continuous operating state in which the fault ocurred.

3.4 Places of Latent Nestling

Latent nesting place are defined by the method of discrete type, confirming that all the faults in the system must be assigned exclusively to the set places $PLNf$. However, in a hybrid

system, if there is a continuous place P_i^c which represents during a certain time t an operating, according to the state or states as a discrete place, the faults are assigned to this continuous place where $PLNf_i \in P^C$. This means that the generated faults by the anomalous behavior of the continuous variable somewhere P_i^C are nestling in the same continuous place now called $PLNf_i^C$, because this hybrid character.
The representation of the continuous behaviour normal marks are the numeric text type, while the faulty marks and discrete behaviour normal marks are the same according to the method proposed in the previous chapter

3.5 Trajectorys of fault verification and fault recovery

These trajectories are defined only by the fault and recovery transitions, adding some restrictions to include the status of the places of normal behavior, and the marks of normal behavior. These restrictions are presented in the status and degree of transitions validation and complexity as for the construction of the fault transitions in continuous places.
Definition 3. A fault or recovery transition in a DHCPN is enable for discrete places if each place $PLNf_k^D$ *or* PVf in ${}^{o}TF_j$ meets the condition:
For Tf:

$$m(PLNf_k) \geq Pre(PLNf_k, Tf_j) \tag{15}$$

For Tr:

$$[m(PVf) \geq Pr\,e(PVf, Tr_j)] \wedge [m(PLNf_k) \geq Pre(PLNf_k, Tr_j)] \tag{16}$$

The possibility of exist continuous variables within the hybrid model, implies the possibility of perform an analysis to obtain new diagnoses on these same variables. The main idea is to use classical techniques of fault diagnosis based on models or based on heuristics, (Isermann, 1997). For example, if be use the technique based on quantitative models that is the residue generation and the subsequent evaluation.
Some techniques used in this area of residue generation are the parity equations, (Gertler, 1998), and observers (Chen & Patton, 1999).
To find the residues its neccesary obtain the dynamic model of operation of the continuous variables (typically in differential equations), and isolate the variables according to obtain a residue. Depending on the complexity can be represented in state variables, as in the hybrid PN analysis, (Demongodin & Koussoulas, 1998).
The idea in this new approach to diagnosis in hybrid systems, consists in obtain in every continuous place a series of residues of the form $r(t) = y(t) - \hat{y}(t)$, being $\hat{y}(t)$ the variable represented by the continuous place, and $y(t)$ the measured variable by the system or process in a real time. The residue is obtained directly in the continuous place, while the residual evaluation is realized in each fault and recovery transition using knowledge expert for defined a signatures fault and isolate the fault occurrence in the place PVf. Every one of possibility theseresidues $(r_i(k))$ will be evaluate respect to the τ_i, set according to previous or heuristic knowledge.
To define a systematic approach of this Latent Nestling Method for Hybrid Systems, is necessary raise some new conditions to the continuous analysis called as operating states OS and fault signatures. This new approach will depend on the analized system as for the continuous places influence into themselves.

3.6 States of hybrid operation

Due to the continuous nature present in the hybrid models, its important to analyze the continuous places influence into themselves according of the system to treat. This influence in continuous places sites is an important factor in an effect known as coupling faults, which involves erroneous readings of faults by propagation of these faults, (García & Correcher, 2006). This factor is used to analyze the residues of continuous places in a more systematic way, and achieve a more effective fault isolation.

For every hybrid system exist three influence types according to the continuous places behaviour. Fig 3. a) Continuous isolated places b) Continuous places cascade influenced c) Continuous places cyclic influence.

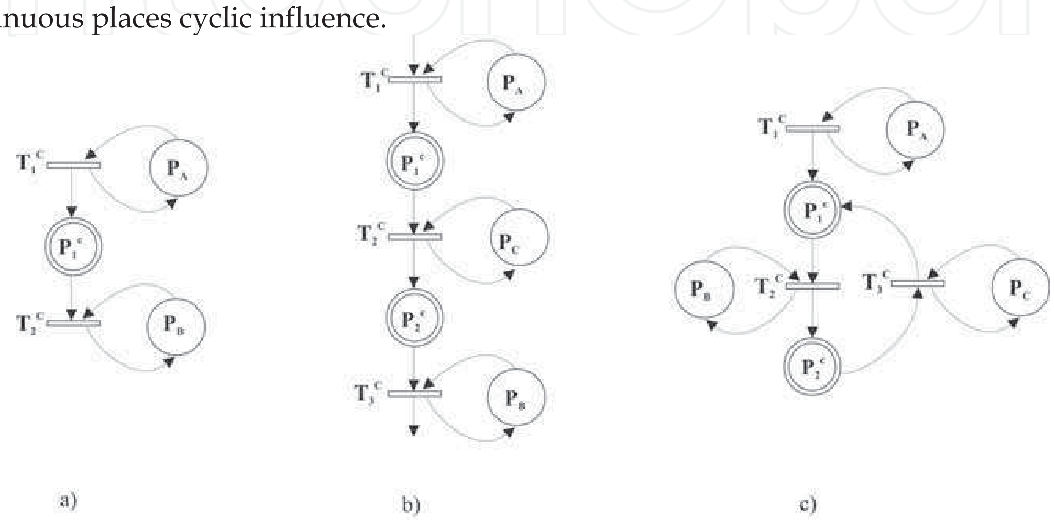

Fig. 3. Types of continuous influence according to the hybrid model

Also, exist an operating states for each continuous place, indicating the behavior of the continuous variable modeled. These operating states depend on the operation of the discrete places that control the continuous place and to include some signatures for the residual faults analysis.

A hybrid model has a set of operating states for the failure and recovery as well:

$$OS = OS_f \cup OS_r \tag{17}$$

Where OS_f are the fault operating states, and OS_r the recovery operating states.

(a) Continuous isolated places

These models usually have only one place continuous in the hybrid model with a single vector of operating states, but depending on the model, may be exist several continuous places not be influence into themselves, which would mean a vector of operating states for each continuous place. Where $OS_f=(os_l,...,os_k)$ being that l and k subscript correspond to the places P_l^c and P_k^c, being $|OS_f|$ in this case the number of P^c isolates in the model. Every os_i is a vector that contain many operating states as signature faults for every contiuous place P_i^c, thus, $os_i=(Sf_1(k),..., Sf_m(k))$.

The OS_r set has the same definition that OS_f, being the vector os_i contains in this case recovery signature faults.

(b) Continuous places mutually influenced

- Cascade influence
  In these models of continuous places are influenced one to one, however the information flow is transferred in an open loop, meaning that the behavior of a continuous place P_i^c directly influences the behavior of the continuous place $P^c{}_{i+1}$, successively, but not influence in the immediately preceding, $P^c{}_{i-1}$.
- Cyclic influence
  These model are characterized by has flow behavior in closed or feedback, meaning that exist a mutually influence in every continuous places according to control of the discrete places.
  Both, continuous places cascade influenced as a continuous places cyclic influence exist in the same manner as for every continuous isolated places a single fault transicion Tf_i, which defines a number of *Pre* arcs for this Tf_i as:

$$\Pr e^{Tf_i} = \sum_{j=x}^{n} (P_j^c \times Tf_i) \qquad (18)$$

  Where, x is the initial continuous place influenced, and n is the last continuous place influenced
- Mixed influence
  This models may have mixed operation structures, as isolates , cascades or cyclic.

To find the operating states of the continuous places mutually influenced greater ease, it is necessary to make a table called "table of continuous places influenced." This table will have the number of operating states of the model. Also, obtain the fault signatures according to the discrete places that influencing each continuous place. This table shows the main influence of a continuous place over the other continuous place according to the discrete places that interact with the continuos places, allowing obtain the fault and recovery transitions.

In every continuous places exist five operating states that depend to the transition enable degree, according to the discrete influence, these are: increase, decrease, mixed or resting.

In table 1 is shown the influenced for the models in figure 3. The X state means that this combination of discrete events its not possible by the control.

a)

P_B	P_A	P_1^c	
0	0	Sf_1	os_1
0	1	Sf_2	os_2
1	0	Sf_3	os_3
1	1	X	

b)

P_A	P_B	P_C	P_1^c	P_2^c	
0	0	0	X	X	
0	0	1	Sf_1	Sf_2	os_1
0	1	0	Sf_3	Sf_4	os_2, os_3
0	1	1	X	X	
1	0	0	Sf_5	Sf_6	os_4, os_5
1	0	1	X	X	
1	1	0	X	X	
1	1	1	X	X	

c)

P_A	P_B	P_C	P_1^c	P_2^c	
0	0	0	X	X	
0	0	1	Sf_1	Sf_2	os_1
0	1	0	Sf_3	Sf_4	os_2
0	1	1	X	X	
1	0	0	Sf_5	Sf_6	os_3, os_4
1	0	1	X	X	
1	1	0	X	X	
1	1	1	X	X	

Table 1. Continuous places mutually influenced

The table 1 a) shown the fault signatures by the behaviour of the continuous place P_1^c, where:

$os_1=(Sf_1(k),Sf_2(k),Sf_3(k))$
Sf_1= Fault signature for resting state
Sf_2= Fault signature for increase state (filling)
Sf_3= Fault signature for decrease state (emptying)
Determining a fault type $<f_i, S_i>$, where each S_i correspond to a fault signature. Analyzing b), is interesting to note that the operating state os_1 is mixed, being that P_1^c influence the behaviour of P_2^c, this influence indicate that the residue must be analized together for a better isolation faults. Likewise, tha fault signatures Sf_3 and Sf_4 are part of different operating states, one place in resting and the other place in decrease. Analyzing c) be shown that the fault signatures Sf_3 and Sf_4 are part of the same operating state, and is necessary analizing its residues together for isolate the faults.
Figure 4 shows the same figure 3 but with fault transitions implemented according to previous analysis.

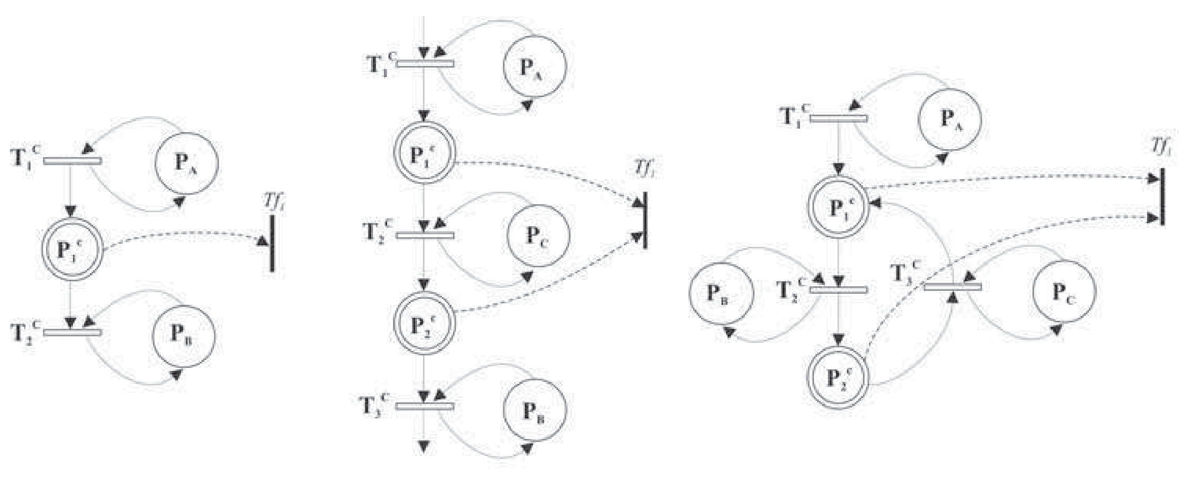

Fig. 4. Fault transitions of continuous places influenced

3.7 Fault signatures

These fault signatures represent the faults in the place PVf isolate of recurrent manner according to the residual behavior like using the threshold for every operating state. For example:

$$Sf_n(k) = \begin{cases} < f_i, s_n > if \rightarrow r_i(k) > \tau_i \\ < f_i f_k, s_n > if \rightarrow r_i(k) < \tau_j \\ \vdots \\ < f_m, s_n > \cdots \end{cases} \tag{19}$$

For this case the detection and isolation of individual faults f_i, or simultaneous faults f_if_k, is determinated by the dynamic conditions in the discrete or continuous marking and consequently in each state reached by the system. Likewise, by the set of not expected

readings from discrete sensors and the signatures faults, according to the current operating state in the continuous state.
As seen in the LNM section, exist a set of not expected values of sensor readings $SROV_{nev}(M(k))$, and a set of expected values of sensor readings $SROV_{ev}(M(k))$ for a given discrete marking, that permit associate the fault verification from latent nestling place *PLNf* to the verification place *PVf*, or otherwise to recovery faults.
Due to the possibility of include faults from the continuous dynamics, the set $SROV_{nev}(M(k))$ will include a fault signature $Sf_n(k)$ in a case of single continuous place, or a os_i fault signatures vector in a mutually influenced places, according to the faults obtained by residues, by the dynamic behavior of the continuous place P_i^c and the place of latent nestling faults *PLNf* that influence this continuous dynamic.
Also, include a recovery signature $Sr_n(k)$ to the normal behavior of the residue, according to the dynamic behavior of the continuous place P_i^c and the latent nestling places $PLNf_n$ that influence this continuous dynamic. The recovery signatures are defined in the same manner as the fault signatures described above, changing the label "*f*" of fault by "*r*" for recovery. For a more compact notation in terms of fault and recovery transitions, the enable "E" for any transitions is given as:
For *Tf*:

$$Ef = SROV_{nev}(M_k),\ Sf_n(k) \tag{20}$$

For *Tr*:

$$Er = SROV_{nev}(M_k),\ Sr_n(k) \tag{21}$$

Finally, to define the fault trajectory traced from a continuous place P_i^c, which contains a faulty mark of this type $<f_i,S_n>$, a continuous mark of normal behaviour $<h>$, verified the discrete state with a behavior normal mark $<\bullet q>$, and is influenced by a residue outside of a designated threshold, would be expressed the form:

$$\begin{gathered}(M(PLNf_k(<\bullet q>)) \wedge M(P_i^c(<h>,(f_i,S_n))))\\ [Tf_j/\ Sf_n(k)(M(PLNf_k(<\bullet q>)) \wedge M(P_i^c(<h>,(f_i,S_n))))\\ > M(PVf((f_i,S_n)))\end{gathered} \tag{22}$$

The figure 5 represents this behaviour

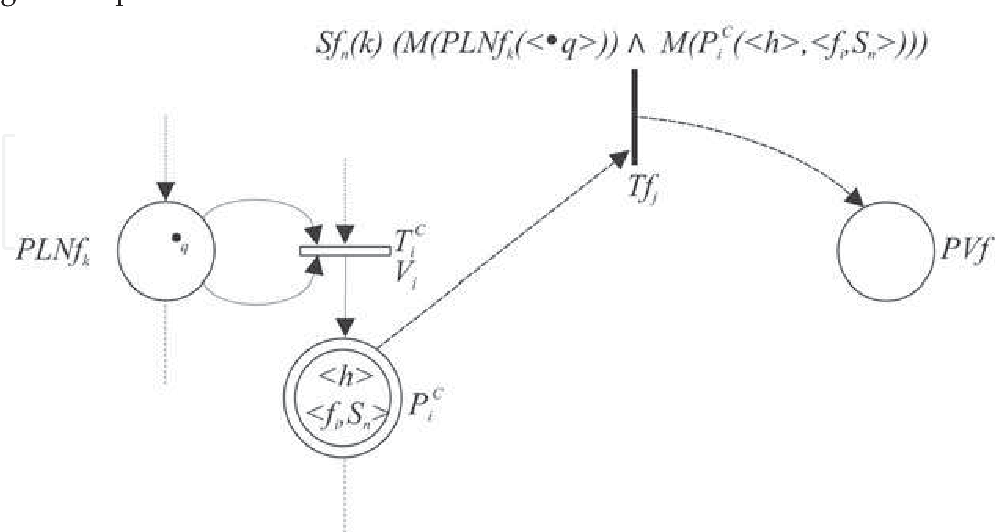

Fig. 5. Trajectory for fault verification (abnormal behavior of the residue in a continuous variable)

Eventually, the same fault (f_i,S_n), can experiment a recovery process to the origin place P_i^c. In this way the model can be receptive to the treatment of intermittent faults. This recovery is expressed as:

$$\begin{gathered} M(PVf((f_i,S_n))) \\ [Tr_j / Sr_n(k)\ (M'(PLNf_k(<\bullet q>))\ \wedge M'(P_i^c(<h>)\ \wedge M(PVf((f_i,S_n)))) \\ > M(P_i^c(<h>,(f_i,S_n))) \end{gathered} \tag{22}$$

The figure 6 represents this behaviour.

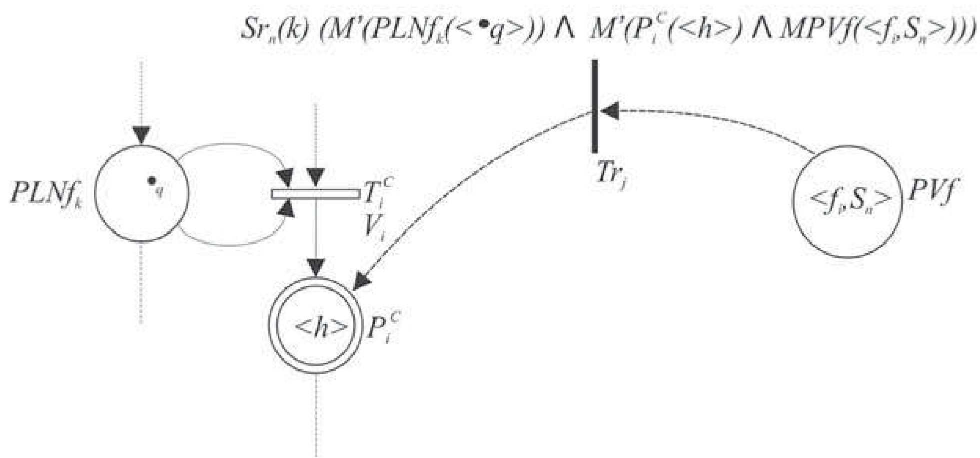

Fig. 6. Trajectory for fault recovery (normal behavior of the residue in a continuous variable)

3.8 Diagnosability of model

The diagnosability concept is maintained according to the previous paragraph, but be must include the fault signatures for each operating state of the continuous places analyzed P_i^c.

$$\begin{gathered} \forall f_i \in f\ \ \exists(M(PLNf_i^c(<h>,<f_i,S_n>))) \\ [Tf_j / Sf_n(k)(M(PLNf_i^c(<h>,<f_i,S_n>))) \\ >M(PVf(<f_i,S_n>)) \end{gathered} \tag{23}$$

Likewise, its necessary satisfy the condition that at least one fault signature $Sf_n(k) \subseteq os_i$ must exist for each continuous place P_i^c.

$$\forall\ PLNf_i^c \in P^c\ \exists\, Sf_n(k) \subseteq os_i \tag{24}$$

3.9 Methodology example

The example system consists of a liquid storage tank, where it has: A storage system or tank, 3 actuators (2 pass valves, 1 mixer), 3 sensors (2 flow binary, 1 level type ultrasound). In Figure 7 shows the physical structure of the system.

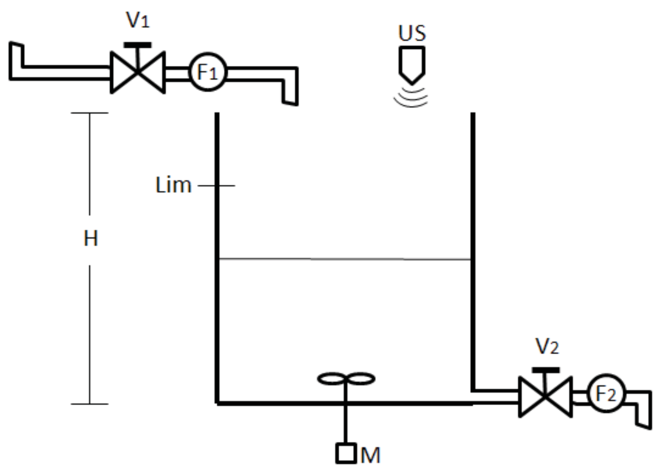

Fig. 7. Example of hybrid system

The process starts giving orders to open valve V_1 for filling tank with a flow ratio 2v.u/t.u (volume units per time unit) until the position $Lim = 30$ indicated in the figure (this position is a level indicator for the discrete measured by ultrasonic sensor). Then, the mixer M is activated during $t_1 = 20$ seconds and close valve V_1 for not deposit more product. Finally, its necessary opening valve V_2 to empty the tank with a flow ratio 3v.u/t.u, and deactivate the mixer. Both input and output flow is a fixed ratio, which indicates that the function of filling and emptying is linear. In the real model is used $K\sqrt{h(t)}$, as an outflow, but in the previous simulation, the flow ratio is indicated above. The process runs on a cyclical mode. In figure 8 can be observed the hybrid model using the Sirphyco software (David & Alla, 2005).

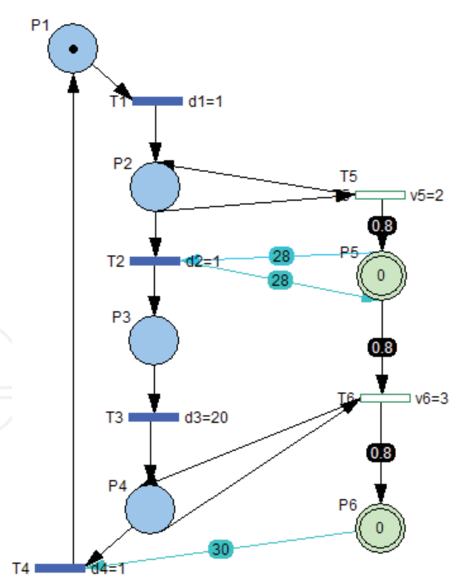

Fig. 8. Hybrid model using Sirphyco tool

To analyze the behavior of discrete dynamic system, will be to obtain four reachable markings o the normal behavior thus:
M_0= initial condition, close valve V_1

M_1= the valve V_1 is ordered to open
M_2= the valve V_1 is ordered to close, and the mixer is activated
M_3= the valve V_2 is ordered to open, and the mixer is deactivated
Getting the initial vector of reachable markings for the system. $^*M_0$= (M_0, M_1, M_2, M_3.)
The dynamic analysis of the continuous system is governed by a simple differential equation.

$$\frac{dh(t)}{dt} = \frac{1}{A}(q_1(t) - q_2(t)) \tag{25}$$

Where A is the area of the tank, $\frac{dh(t)}{dt}$ the height variation respect to the time, $q1$ and $q2$ the input and output flow for the valves V_1 and V_2 respectively.
To define the faults is used to knowledge of the proposed system.
Where the faults can be: stuck valves fault, leakage in the tank fault, sensors fault. Classifieds these faults in identifiers using coloured marks, we have:
f_1= stuck open valve V_1 fault
f_2= stuck close valve V_1 fault
f_3= stuck open valve V_2 fault
f_4= stuck close valve V_2 fault
f_5= leakage in the tank fault
f_6= level sensor fault
The readings of discrete behavior sensor are:
$srov_{11}(M_k)=\{F_1,NF_1\}$
$srov_{12}(M_k)=\{F_2,NF_2\}$
Using the level sensor as a measured discrete
$srov_2(M_k)=\{L,NL\}$
Where L= Exist a level and NL= the tank is empty.
In table 2 shown the faults according the sensors readings and discrete marking

F_1	F_2	L	M_0	M_1	M_2	M_3
0	0	0	$SROV_{ev}$	f_2	f_6	f_4f_6
0	0	1	f_6	f_6f_2	$SROV_{ev}$	f_4
0	1	0	f_3f_6	$f_2f_3f_6$	f_3f_6	f_6
0	1	1	f_3	f_2f_3	f_3	$SROV_{ev}$
1	0	0	f_1f_6	f_6	f_1f_6	$f_1f_4f_6$
1	0	1	f_1	$SROV_{ev}$	f_1	f_1f_4
1	1	0	$f_1f_3f_6$	f_3f_6	$f_1f_3f_6$	f_1f_6
1	1	1	f_1f_3	f_3	f_1f_3	f_1

Table 2. Fault behaviour according the discrete sensor readings

For continuous analysis must be examine continuous variables that influence the process (in this case one variable) and discrete sensors that influence this continuous place. For this case there is a single continuous place of isolated type, implying that exist a single operating state and a series of fault signatures for each discrete place that influence the behavior of the continuous place.

OSf=*osf*$_5$. The operating state *osf*$_5$ corresponds to the vector that contain the number of operating states of the continuous place $P_5{}^c$ identified with a fault signature for each fault as follows:
$mf_5=(Sf_2(k),Sf_3(k),Sf_4(k))$, known that $PLNf_2$, $PLNf_3$, $PLNf_4$ are the places that influence the behavior of the continuous place.
Due to the presence of the continuous type sensor for the height measured, is possible a comparison between the measured height *h* and the estimated height *h'*.
Using the equation 25:
Case 1: first operating state, increase "filling"

$$h' = \frac{1}{A}\int q_1(t)\cdot dt \tag{26}$$

Where the residue r_1 is obtained:
$r_1=h-h'$

$$Sf_2(k) = \begin{Bmatrix} \langle f_5, S_2\rangle if \rightarrow r_1 > \tau_{11} = 0.3 \\ \langle f_6, S_2\rangle if \rightarrow r_1 < \tau_{12} = -1 \end{Bmatrix}$$

$$Sr_2(k) = \begin{Bmatrix} \langle f_5, S_2\rangle if \rightarrow r_1 > \tau_{41} = 0.15 \\ \langle f_6, S_2\rangle if \rightarrow r_1 < \tau_{42} = -0.5 \end{Bmatrix}$$

Case 2: second operating state "resting"
Where the height is a constant and the residue r_2 is obtained:
$r_2=h-h'$

$$Sf_3(k) = \begin{Bmatrix} \langle f_5, S_3\rangle if \rightarrow r_2 > \tau_{21} = 0.1 \\ \langle f_6, S_3\rangle if \rightarrow r_2 < \tau_{22} = -0.5 \end{Bmatrix}$$

$$Sr_3(k) = \begin{Bmatrix} \langle f_5, S_3\rangle if \rightarrow r_2 > \tau_{51} = 0.08 \\ \langle f_6, S_3\rangle if \rightarrow r_2 < \tau_{52} = -0.4 \end{Bmatrix}$$

Case 3: third operating state, decrease "emptying"

$$h' = \frac{1}{A}\int q_2(t)\cdot dt \tag{27}$$

Where the residue r_3 is obtained:
$r_3=h-h'$

$$Sf_4(k) = \begin{Bmatrix} \langle f_5, S_4\rangle if \rightarrow r_3 > \tau_{31} = 1 \\ \langle f_6, S_4\rangle if \rightarrow r_3 < \tau_{32} = -0.4 \end{Bmatrix}$$

$$Sr_4(k) = \begin{Bmatrix} \langle f_5, S_4\rangle if \rightarrow r_3 > \tau_{61} = 0.8 \\ \langle f_6, S_4\rangle if \rightarrow r_3 < \tau_{62} = -0.3 \end{Bmatrix}$$

The thresholds set $\tau = (\tau_{11},...,\tau_{32})$ are given by knowledge expert and it analyze is according to different factors as: hysteresis, disturbances, noise, as well as the sensor sensitivity and sensor resolution.
Just as there are a fault operating states for each continuous place, too there are a recovery operating states. In these recovery states the τ values changes because the sensor hysteresis. For example, if $r_1=0.4$ when the process is filling, the isolate and recovery fault f_5 are given by the expression:
The fault isolation f_5 in this condition occurs if:

$$(M(PLNf_2(<\bullet n>)) \wedge M(P_5{}^c(<h>,\langle f_5,S_2\rangle)))$$
$$[Tf_3 / r_1>0.3\ (M(PLNf_2(<\bullet n>))\ \wedge M(P_5{}^c(<h>,\langle f_5,S_2\rangle)))$$
$$> M(PVf(\langle f_5,S_2\rangle))$$

The fault recovery f_5 in this condition occurs if:

$$M(PVf(\langle f_5,S_2\rangle))$$
$$[Tr_3 / r_1<0.15\ \ (M'(PLNf_2(<\bullet n>))\ \wedge M'(P_5{}^c(<h>)\ \ \wedge M(PVf(\langle f_5,S_2\rangle)))$$
$$> M(P_5{}^c(<h>,\langle f_5,S_2\rangle))$$

At it is observed , the diagnosis system is able to detect the isolate fault of individual type f_1, f_2, f_3, f_4, f_6, and simultaneous type f_1f_6, f_1f_3, f_1f_4, $f_1f_3f_6$, $f_1f_4f_6$, f_2f_3, $f_2f_3f_6$, f_4f_6, f_6f_2, as well as process fault of type $\langle f_5,S_2\rangle$, $\langle f_6,S_2\rangle$, $\langle f_5,S_3\rangle$, $\langle f_6,S_3\rangle$, $\langle f_5,S_4\rangle$, $\langle f_6,S_4\rangle$.
In figure 9 be shown the final model for the tank example

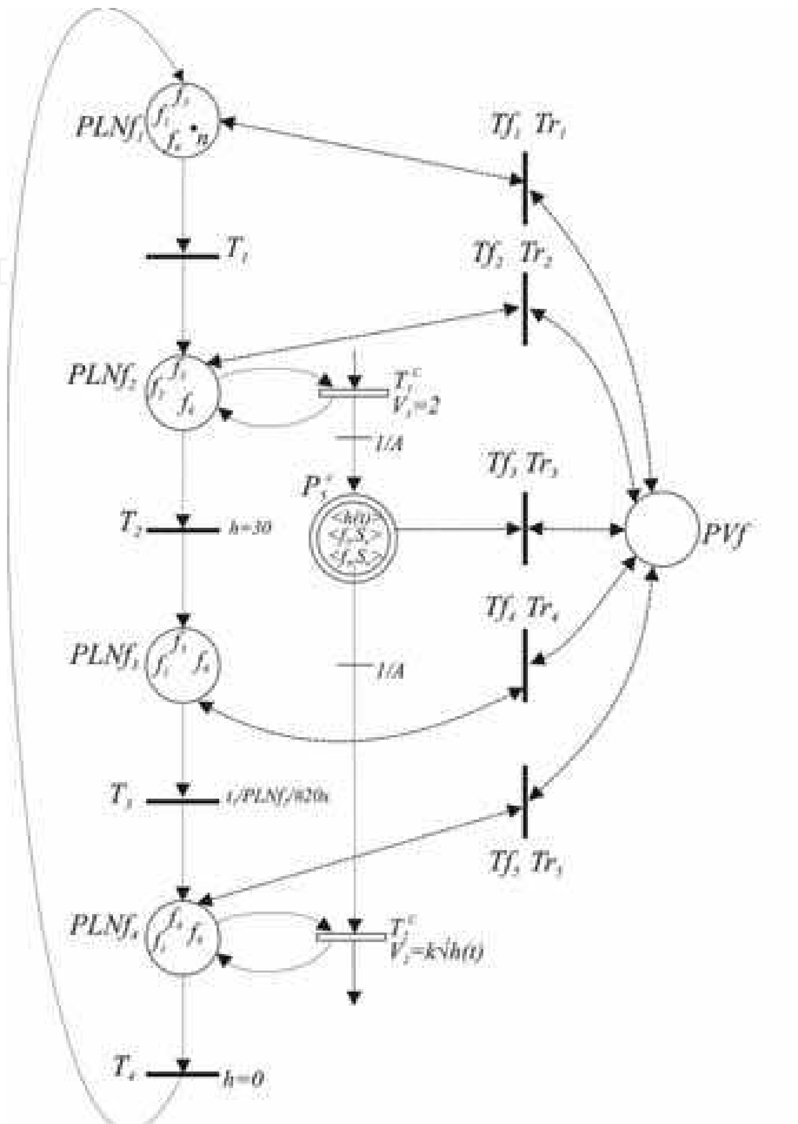

Fig. 9. DHCPN model (example of filling tank)

4. Conclusions

The method shows the reduction and simplicity of the system models are discrete, continuous or hybrid, giving them characteristics of readability, implementability treatability and no matter how many sensors to treat or how many faults to diagnose; imposible features to obtain with other methodologies such as MEFs.

The hybrid nestling technique shows the need to analyze the residues with the information of the discrete state in normal behavior for characterize the type of fault, its location and subsequent isolation.

Operating states, and the influence tables of continuous places, offer an overview of the system's behavior as sharing your information, being this information a continuous variable to treat in the model. This overview provides the possibility of locating the fault transitions thus analyze the fault coupling, to avoid false warnings in the verification place.

The Latent Nestling Methodology focused in continuous and hybrid systems presents an excellent and clear solution to fulfill the objectives of diagnosis and isolation for any faults type that may arise in the system.

5. References

Caselitz, P.; Giebhardt, J.; Mevenkamp, M. Development of a Fault Detection System for Wind Energy Converters, *Proceedings of the EUWEC '96*, Göteborg, pp. 1004 – 1007.

Chen, J. & Patton, R. J. Robust model-based fault diagnosis for dynamic systems *Kluwer Academic Publishers,* 1999.

Correcher, A.; García, E.; Morant, F.; Quiles, E. & Blasco. R. Intermittent Failure Diagnosis based on discrete event models, *Proceeding of 7'Th Workshop On Discrete Event Systems WODES04*, pp. 151-157, 2004.

David, R. & Alla, H. Discrete, Continuous, and Hybrid Petri Nets*, Springer-Verlag*, Berlin. 2005.

Demongodin, I. & Koussoulas, N. Differential Petri nets: Representing Continuous Systems in a Discrete Event World, *IEEE transactions on Automatic Control.* 1998*, 38*, pp. 573-579

García, E.; Morant, F.; Blasco, R.; Correcher, A.; Quiles, E. Centralized Modular Diagnosis and the Phenomenon of Coupling, *Workshop On Discrete Event Systems WODES02,* Zaragoza, Spain October 2002.

García, E.; Rodríguez., L.; Morant., F.; Correcher., A.; Quiles., E. & Blasco, R. Fault Diagnosis with Coloured Petri Nets Using Latent Nestling Method, *Proceedings of ISIE08 Cambridge UK,* 2008.

García, E.; Rodríguez., L.; Morant., F.; Correcher., A. & Quiles., E. Latent Nestling Method: A new fault diagnosis methodology for complex systems *IECON08, The 34th Annual Conference of the IEEE Industrial Electronics Society, Orlando, Florida, U.S.A,* 2008.

Gertler, J. Fault Detection and Diagnosis in Engineering Systems, *Marcel Dekker,* 1998.

Isermann, R. Supervision, fault-detection and fault-diagnosis methods -- An introduction *Control Engineering Practice,* 1997*, 5*, pp. 639 - 652

Jensen, K. Coloured Petri Nets 1: Basic Concepts, Analysis Methods and Practical Use, *Springer-Verlag,* 1992 Berlin, Germany.

Keller, W. Petri Nets for Reverse Engineering, *Universidad de Zurich,* 2000.

Patton, R. J.; Lopez-Toribio, C. J. & Uppal, F. J. Artificial intelligence approaches to fault diagnosis for dynamic systems, *International Journal of Applied Mathematics and Computer Science,* 1999*, 9(3)*, pp. 471-518.

Rodríguez., L.; García., E.; Morant., F.; Correcher., A. & Quiles, E. Application of Latent Nestling Method using Coloured Petri Nets for the Fault Diagnosis in the Wind Turbine Subsets *Proceedings of ETFA'08, Hamburg, Germany,* 2008.

Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K. & Teneketzis, D. Failure diagnosis using discrete-event models, *Control Systems Technology IEEE Transactions on*, 1996, 4, pp. 105-124.

Petri Nets Applications

Edited by Pawel Pawlewski

ISBN 978-953-307-047-6

Hard cover, 752 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

Petri Nets are graphical and mathematical tool used in many different science domains. Their characteristic features are the intuitive graphical modeling language and advanced formal analysis method. The concurrence of performed actions is the natural phenomenon due to which Petri Nets are perceived as mathematical tool for modeling concurrent systems. The nets whose model was extended with the time model can be applied in modeling real-time systems. Petri Nets were introduced in the doctoral dissertation by K.A. Petri, titled "„Kommunikation mit Automaten" and published in 1962 by University of Bonn. During more than 40 years of development of this theory, many different classes were formed and the scope of applications was extended. Depending on particular needs, the net definition was changed and adjusted to the considered problem. The unusual "flexibility" of this theory makes it possible to introduce all these modifications. Owing to varied currently known net classes, it is relatively easy to find a proper class for the specific application. The present monograph shows the whole spectrum of Petri Nets applications, from classic applications (to which the theory is specially dedicated) like computer science and control systems, through fault diagnosis, manufacturing, power systems, traffic systems, transport and down to Web applications. At the same time, the publication describes the diversity of investigations performed with use of Petri Nets in science centers all over the world.