Conciencia Tecnológica No. 37, Enero - Junio 2009

Reduced Petri Net Diagnosers for Detecting and Locating Faults

Research

Dra. Elvia Ruiz Beltrán[1] and Dr. Jorge Luis Orozco Mora[2]
[1]Department of Systems and Computer, 2Department of Electrical-Electronic Engineering
Instituto Tecnológico de Aguascalientes. Av. Adolfo López Mateos No. 1801 Ote. Fracc. Bona Gens.
Aguascalientes, Aguascalientes. México.
Phone/Fax +(52 449) 910-5002 Ext. 106 E-mail: eruiz@ita.mx, drorozco@ita.mx

Abstract

This paper deals with fault detection and location of Discrete Event Systems (DES) modeled using Interpreted Petri nets (IPN). In this paper two efficient methods for obtaining reduced diagnosers IPN models are proposed: based on the DES model, a Petri net is synthesized depending on each methodology; each net consists of a single place (or more) and the same number of transitions that the system model has; the current marking of this place (places) is enough to determine and locate faults occurring within the DES.

Keywords: Discrete Event Systems, Interpreted Petri Nets, Fault detection and location.

Resumen

Este artículo se relaciona con la detección y localización de fallas de Sistemas de Eventos Discretos (SED), los cuales son modelados mediante Redes de Petri Interpretadas (RPI). En este artículo se proponen dos métodos eficientes para obtener modelos diagnosticadores reducidos de RPI: basándose en el modelo del SED, una red de Petri se sintetiza dependiendo de cada metodología; cada red diagnosticador0 se compone de un lugar (o más) y el mismo número de transiciones que el modelo del sistema; el marcado actual del lugar (o lugares) es suficiente para detectar y localizar las fallas que se presenten en el SED.

Palabras clave: Sistemas de Eventos Discretos, Redes de Petri Interpretadas.

Introduction

The analysis of the diagnosability property of a DES has been addressed through several approaches and methods, namely that based on artificial intelligence techniques and that based on discrete event models. Recently, finite automata (FA) and Petri nets (PN) have been widely used as modeling formalisms and formal tools for fault diagnosis.

In [1] M. Sampath et al. introduced the notion of diagnosability and proposed a method for designing an on-line diagnoser for determining the diagnosability property.

In [4], [5] and [7] Prock, Genc and Hadjicostis respectively proposed methods based on PN models to detect and isolate the faults presented in the system. In [4] the tokens residing in P-semiflows are monitored and, depending on the quantity of these tokens, faults into the system are determined. In [5] the held approach analyzes the reachability graph to isolate faults, leading to NP-complete algorithms. The strategy presented in [7] consists in adding one more place to the system model, the marking of this place can be used to determine and isolate system faults; in this approach however, it is assumed that all places are measurable, that all transitions are controllable, and that system and observer can be synchronized each other. These conditions are hardly fulfilled in current systems.

Based on [7] and the structural characterization of diagnosability [9], we propose a novel diagnosis scheme that use simple and reduced IPN models for monitoring the DES behavior. The monitoring models play a similar role as used in [7], however they can operate using only partial information on the marking (measurable places), and their synchronization with the DES model are not longer needed.

This work is organized as follows: in preliminaries section basic definitions of IPN are provided. Later, it is presented a procedure to obtain an IPN model. Next, two methodologies for structuring the diagnosis scheme are proposed. Then, it is included an example for illustrating the notions herein introduced. Finally conclusions are given.

Preliminaries

This section presents the basic concepts and notation of PN and IPN used in this paper.

Definition 1: A Petri Net structure G is a bipartite digraph represented by the 4-tuple G=(P,T,I,O) where:

- P = {p1,p2,...,pn} and T ={t1,t2,...,tm} are finite sets of vertices named places and transitions respectively,

- I (O) : $P \times T \rightarrow Z^+$ is a function representing the weighted arcs going from places to transitions (transitions to places); Z^+ is the set of nonnegative integers.

Pictorially, places are represented by circles, transitions are represented by rectangles, and arcs are depicted as arrows. The symbol $^\bullet t_j$ ($t_j^\bullet$) denotes the set of all places p_i such that $I(p_i,t_j)\neq 0$ ($O(p_i,t_j)\neq 0$). Analogously, $^\bullet p_i$ ($p_i^\bullet$) denotes the set of all transitions t_j such that $O(p_i,t_j)\neq 0$ ($I(p_i,t_j)\neq 0$).

The pre-incidence matrix of G is $C^- = [c_{ij}^-]$ where $c_{ij}^- = I(p_i,t_j)$; the post-incidence matrix of G is $C^+ = [c_{ij}^+]$, where $c_{ij}^+ = O(p_i,t_j)$; the incidence matrix of G is $C = C^+ - C^-$.

A marking function M : $P \rightarrow Z^+$ represents the number of tokens (depicted as dots) residing inside each place. The marking of a PN is usually expressed as an n-entry vector.

Definition 2: A Petri Net system or Petri Net (PN) is the pair $N=(G,M_0)$, where G is a PN structure and M_0 is an initial token distribution.

In a PN system, a transition t_j is enabled at marking M_k if $\forall p_i \in P$, $M_k(p_i) \geq I(p_i,t_j)$; an enabled transition t_j can be fired reaching a new marking M_{k+1} which can be computed as $M_{k+1} = M_k + Cv_k$, where $v_k(i)=0$, $i\neq j$, $v_k(j)=1$, this equation is called the PN state equation. The reachability set of a PN is the set of all possible reachable marking from M_0 firing only enabled transitions; this set is denoted by $R(G,M_0)$.

This work uses Interpreted Petri Nets (IPN) [8] an extension to PN that allow associating input and output signals to PN models.

Definition 3: An IPN (Q, M_0) is an IPN Interpreted Petri Net structure $Q=(G,\Sigma,\lambda,\varphi)$ with an initial marking M_0.

- G is a PN structure
- $\Sigma = \{\alpha_1,\alpha_2,...,\alpha_r\}$ is the alphabet of input symbols α_i.
- $\lambda : T\rightarrow\Sigma\cup\{\varepsilon\}$ is a labeling function of transitions with the following constraint: $\forall t_j,t_k \in T$, $j\neq k$, if $\forall p_i$ $I(p_i,t_j) = I(p_i,t_k) \neq 0$ and both $\lambda(t_j) \neq \varepsilon$, $\lambda(t_k) \neq \varepsilon$, then $\lambda(t_j) \neq \lambda(t_k)$; ε represents a system internal event.
- $\varphi : R(Q,M_0)\rightarrow (Z^+)^q$ is an output function, that associates to each marking in $R(Q,M_0)$ q-entry output vector; q is the number of outputs.

In this work φ is a q×n matrix. Each column of φ is an elementary or null vector. If the output symbol i is present (turned on) every time that $M(p_j)\geq 1$, then $\varphi(i,j)=1$, otherwise $\varphi(i,j)=0$.

A transition $t_j \in T$ of an IPN fires it is enabled at marking M_k, and a) If $\lambda(t_j) = a_i \neq \varepsilon$ is provided to the system, or b) If $\lambda(t_j) = \varepsilon$ and t_j is enabled then t_j can be fired. When an enabled transition t_j is fired in a marking M_k, then a new marking M_{k+1} is reached. This fact is represented as $M_k \xrightarrow{t_j} M_{k+1}$; M_{k+1} can be computed using the state equation:

$$M_{k+1} = M_k + Cv_k: \quad y_k = \varphi(M_k) \qquad (1)$$

where C and v_k are defined as in PN and $y_k \in (Z^+)^q$ is the k-th output vector of the IPN.

According to functions λ and φ, transitions and places of an IPN (Q,M_0) can be classified as follows.

Definition 4: If $\lambda(t_i)\neq\varepsilon$ the transition t_i is said to be manipulated. Otherwise it is nonmanipulated. A place $p_i\in P$ is said to be measurable if the i-th column vector of φ is not null, i.e. $\varphi(\bullet,i)\neq 0$. Otherwise it is nonmeasurable.

Definition 5: An IPN $(Q,M_0) = (N,\Sigma,\Phi,\lambda,D,\varphi)$ described by the state equation (1) is event-detectable iff the firing of any pair of transition t_i, t_j of (Q,M_0) can be distinguished from each other by the observation of the sequences of input-output symbols.
The following lemma [8] gives a polynomial characterization of event-detectable IPN.

Lemma 6: Let $(Q,M_0) = (N,\Sigma,\Phi,\lambda,D, \varphi)$ be an IPN described by the state equation (1). (Q,M_0) is event-detectable iff all φC columns are not null and different from each other.

Building IPN Models

We deal with IPN models representing normal and faulty events and states. In a DES an internal event representing a fault is associated to a transition that leads to a place that represents the failed operation. Below is included the procedure followed in the construction of an IPN model.

Procedure 7: Building an IPN model (Q, M_0)
1. Build an IPN model of the normal behavior of the system (Q^N, M_0^N), i.e. when no failures are considered. The set of places of this model is named P^N (normal places) and the set of transitions of this model is

named T^N (normal transitions). Define the initial marking M_0.
2. Define a set of possible failure states F^S. For every state in F^S build another set of places, named P^F.
3. Connect (Q^N, M_0^N) to the places of P^F through new transitions representing the faults (T^F).
4. The matrix φ is extended; one row is added for each $p_i^F \in P^F$; where $\varphi(p_i^N, \bullet) = \varphi(p_j^F, \bullet)$ if place p_i^N is connected to p_j^F through a unique transition in T^F.

Diagnoser Design

We present an extension to the diagnoser proposed in [7].

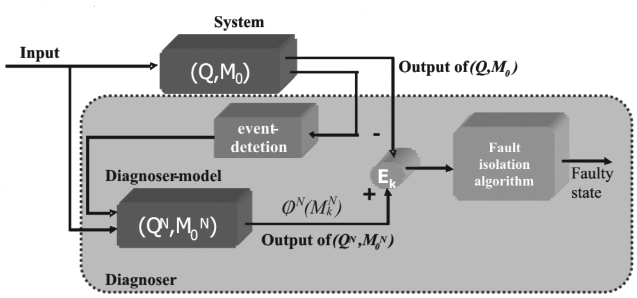

Fig. 1. Scheme for diagnosis

The features considered in this work are:
1. Some places are measurable (not all of them).
2. The diagnoser model is not synchronized with the system; rather the diagnoser model tracks the system.
3. A transition symbol is activated into the system when the transition is enabled or it was disabled by the firing of a faulty transition.
4. The diagnoser is robust and never fails.

These considerations are more realistic than those considered in all previous works since actual systems do not measure all the state variables; furthermore, true synchronization of the diagnoser with the system cannot be carried out in the general case and the diagnoser does not know the fault type within the system (it determines the fault type).

The diagnoser herein proposed consists of four modules: a diagnoser model (DM), an error computation algorithm, a fault isolation algorithm and an event-detection module. The diagnoser scheme is shown in figure 1; its components are defined below.

Diagnoser Model with one place

Definition 8: The proposed diagnoser model structure for the system normal behavior (Q^N, M_0^N) is an IPN (Q^d, M_0^d) where the set of places $P^d = \{p_d\}$ and the set of transitions is $T^d = T^N$, the incidence matrix C^d of (Q^d, M_0^d) is the following

$$C^d = B^T \varphi^N C^N \qquad (2)$$

where C^N is the incidence matrix of (Q^N, M_0^N), φ^N is the output function of (Q^N, M_0^N) and B is a $q\times 1$ non negative vector (q is the number of measurable places of (Q^N, M_0^N)) matrix with nonnegative entries.

The matrix B is computed as follows:

Algorithm 9: Building B

Inputs: C-incidence matrix of an IPN,
q - number of measurable places in the IPN,
Outputs: The matrix B

1. The "base number" b should be computed. In this case $b=2\max(\text{abs}(c_{ij}))+1$, where c_{ij} is an element of incidence matrix C.
2. Define a $q\times 1$ vector.
3. $\begin{bmatrix} b^0 & b^1 & \cdots & b^{q-1} \end{bmatrix}$

This procedure computes matrix B.;

According to the way in which B was constructed, all columns of C^d will be different from zero and different from each other.

The initial marking of the diagnoser model structure is computed as:

$$M_0^d = B^T \varphi(M_0^N) \qquad (3)$$

Error Computation of the DM with one place

Definition 10: Error computation. The k-th error is computed by the following equation:

$$e_k = M_k^d - B^T(\varphi M_k) \qquad (4)$$

Notice that e_k is computed from the diagnoser-model output and not from the marking M_k. It means that the proposed diagnoser is using the system output and not internal system signals (those signals that are non measurable).

Diagnoser Model with more than place

When there is a considerable quantity of tokens contained into the place of the diagnoser model, then a diagnoser model with more than one place is proposed in order to reduce the number of tokens contained into it.

For a diagnoser-model with more than one place, the incidence matrix C^d of (Q^d, Mo^d) is computed using the following equation:

$$C^d = B\varphi^N C^N \quad (5)$$

where B is a $l \times q$ matrix, where $l \leq q$.

The matrix B for this diagnoser model is obtained using the following algorithm.

Algorithm 11: Building matrix B

Inputs: C-incidence matrix of an IPN,
l - number of places in the diagnoser-model,
q - number of measurable places in the IPN,
Outputs: The matrix B

1. The "base number" b should be computed. In this case $b=2\max(\text{abs}(c_{ij}))+1$, where c_{ij} is an element of incidence matrix C.
2. Define a matrix B with $l \times q$, where $l \leq q$.
3. $$B=\begin{bmatrix} b^0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & b^0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & b^0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & b^0 & b^1 & b^2 & \cdots & b^{q-l} \end{bmatrix}$$

This procedure computes the matrix B.;

According to the way in which B was constructed, all columns of C^d must be different from zero and different from each other.

The initial marking for this diagnoser model is computed as:

$$M_0{}^d = B\varphi(M_0{}^N) \quad (6)$$

Error Computation of the DM with one place

$$e_k = M_k{}^d - B(\varphi M_k) \quad (7)$$

Firing Rules of the diagnoser model with one or more places

If a transition $t_i \in T - (T^R \cup T^F)$ is fired in (Q,M_0) then it is fired in $(Q^d, M_0{}^d)$ (it is possible since these transitions are event

The event-detection module determines which transition is fired into the system model and orders that this transition must be fired into the diagnoser model.

Fault Isolation Algorithm for the DM's

Definition 12: Fault isolation. When $e_k \neq 0$, an error is detected, then a faulty marking was reached. The mechanism used to find out the faulty marking is named fault isolation. This work proposes the following algorithm to accomplish this task.

Algorithm 13: Fault isolation

Inputs: M_k, $M_k{}^d$, e_k.
Outputs: p(faulty place), M_f(faulty marking)

Constants: C^d is the IPN diagnoser structure incidence matrix
i = index of the column of C^d such that $C^d(1,i) = e_k$
- $\forall p \in {}^\cdot t_i$, $M_k(p)=0$
- $\forall p \in t_i{}^\cdot$, $M_k(p)=0$
- $\forall p^F \in ({}^\bullet t_i)^{\bullet\bullet} \cap P^F$, $M_k(p^F)=1$
- $M_f = M_k$
- Return (p, M_f)

Definition 14: Let (Q,M_0) be an input-output diagnosable IPN. The 3-tuple (N_B, e_k, A), where
· $N_B = (Q^d, M_0{}^d)$ is the diagnoser structure of (Q,M_0),
· e_k is the error produced by error computation,
· A is the algorithm Fault isolation,
is named an input-output diagnoser for (Q,M_0).

We will prove that after the firing of a finite sequence, the input-output diagnoser for (Q,M_0) detects when a place $p_i \in P^F$ is marked, i.e., it isolates the faulty state.

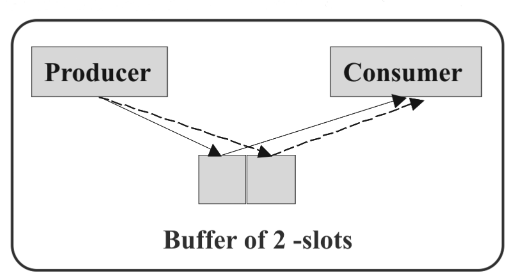

Fig. 2 Producer-Consumer with buffer of 2-slots scheme.

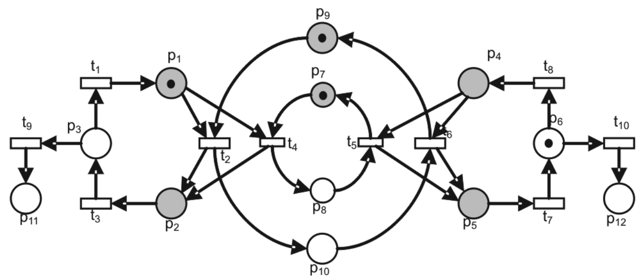

Fig. 3 IPN of the producer-consumer scheme

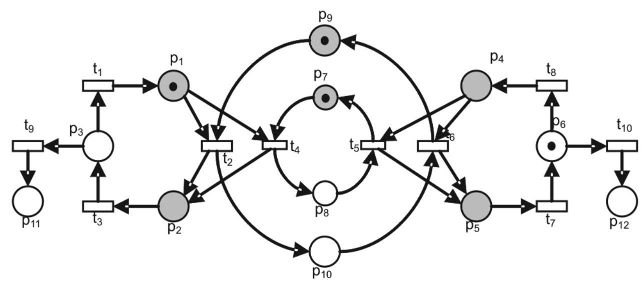

Fig. 4 Normal behavior of the IPN of figure 3.

Example

Consider the producer-consumer scheme depicted in figure 2. The model consists of a producer unit (PU), a consumer unit (CU) and a buffer of 2-slots. The behavior of this system is the following. The producer unit PU creates and delivers products into the free buffer positions. The consumer unit CU retrieves products from the buffer when there is a product stored into a buffer slot. The producer unit PU could reach a faulty state from its producing state. Similarly, the consuming unit could reach a faulty state from its consuming state. Then the places p1, p2, p3 represent the normal behavior of PU and p10 represents the faulty behavior. Places p4, p5, p6 represent the normal behavior of the CU and p12 represents the faulty behavior. The places p7, p8, p9 and p10 represent the 2- slots of the buffer.

The *IPN* (obtained with procedure 7) depicted in figure 3 represents the behavior of the producer-consumer system. Since this *IPN* is input-output diagnosable [9], then a diagnoser can be built for this system. In this case we will use the structure presented in previous section. The normal behavior of this *IPN* is depicted in figure 4; its incidence matrix and output function are:

$$C^N = \begin{bmatrix} 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix};\varphi = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The base obtained to compute B is b=2*1+1=3; since we build B using algorithm 9. We obtain the following vector:

$$B^T = \begin{bmatrix} 1 & 3 & 9 & 27 \end{bmatrix}^T$$

Therefore C^d is:

$$C^d = \begin{bmatrix} -1 & 27 & 1 & 9 & -9 & -27 & 3 & -3 \end{bmatrix}$$

Hence, its associated IPN is depicted in figure 5.

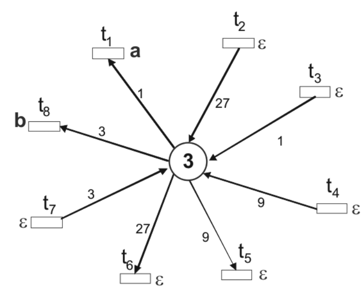

Fig. 5 The IPN diagnoser-model

The initial marking of the diagnoser is M_0^d =[3]. In order to show how the diagnoser works, assume that the following sequence t_2t_3 is executed into the system, then this sequence is fired in the diagnoser. Thus the system output is $(\varphi(M_k))^T = \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix}$, and the marking of the IPN diagnoser is M_k^d =[3 1]. Then e_k= M_k^d - $(B^T)(\varphi M_k)$ = [31] – [31] = 0, thus the system is in a normal state. Now if the faulty transition t_9 is fired, then p_{11} is marked, however no change in the output system is detected. If the symbol of t_1 ($\lambda(t_1)$=b) is given as input to the IPN of the system model and IPN diagnoser, then the diagnoser evolves, and M_{k+1}^d =[30]. Then e_k = −1 indicating the existence of an error. The fault isolation algorithm (algorithm 14) detects that the column 1 of C^d is equal to e_k, thus t_1 was not fired in the system. Then the same algorithm detects the faulty marking and determines that the faulty place p_{11} is marked.

If it is decided to compute a diagnoser model with more than one place (*l=3*), then the matrix *B* for this diagnoser will be the following:

$$B=\begin{bmatrix} b^0 & 0 & 0 & 0 \\ 0 & b^0 & 0 & 0 \\ 0 & 0 & b^0 & b^1 \end{bmatrix} \quad \text{or} \quad B=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

C^d *is:*

$$C^d=\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 3 & 0 & 1 & -1 & -3 & 0 & 0 \end{bmatrix}$$

The corresponded dignoser model is depicted in figure 6, where its initial marking is: $M_0=[0 \quad 1 \quad 0]^T$

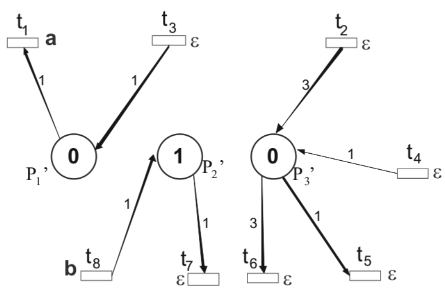

Fig. 6 The IPN diagnoser-model with three places.

It is easy to see that the diagnoser model almost uses 4 tokens into its places. Nevertheless, the diagnoser model (with one place) reaches almost 31 tokens. Thus, the number of tokens is considerably reduced in the diagnoser model with more than one place. But, this diagnoser needs more places for monitoring the system than the diagnoser model with one place needs.

Conclusions

We presented a diagnosis scheme allowing detecting and locating faults of partially observed DES. The diagnosability of the system implies the existence of a monitoring model; then two methods to conceive such models are proposed. Due to the simplicity of the monitoring IPN, the procedure for fault detection and isolation can be efficiently performed. Current research addresses the analysis of a methodology that reduces the potency that the base b is powered.

References

[1] Sampath, M., Sengupta, Lafortune, R., S., Sinnamohideen, K., and Teneketzis D., (1995) "Diagnosability of discrete event systems," *IEEE Trans. Automat, Contr.,* vol 4, no. 9, pp. 1555-1575.

[2] Murata, T., (1989) "Petri nets: Properties, analysis and applications," in *Proc. of the IEEE,* vol. 77, no. 4, pp. 541-580.

[3] Desel J. and Esparza J., (1995), *Free Choice Petri* Nets, Cambridge University Press.

[4] Prock, J., (1991), "A New Technique for Fault Detection Using Petri Nets," *Automatica,* vol. 27, pp. 239-245.

[5] Genc, S., Lafortune, S., (2003), "Distributed Diagnosis of Discrete Event Systems Using Petri Nets," *in Proc. of the 24th ATPN,* pp. 316-336.

[6] Ramírez-Treviño, A., Ruiz-Beltrán, E., Rivera-Rangel, I., López-Mellado, E., (2004), "Diagnosability of Discrete Event Systems. A Petri Net Based Approach," *in Proc. of the IEEE ICRA,* , pp. 541-546.

[7] Ruiz-Beltrán, E., López-Mellado, E., Ramírez-Treviño, A., (2006), "Fault Diagnosis based on Petri net reduced models", in the 3rd International Conference on *Electrical and Electronics Engineering and XII Conference on Electrical Engineering* (Veracruz, Veracruz, México del 6-8 Septiembre), pp. 222-226.

[8] Ramírez-Treviño, A., Rivera-Rangel, I., López-Mellado, E., (2003) "Observability of Discrete Event Systems Modeled by Interpreted Petri Nets," *IEEE Transactions on Robotics and Automation,* vol 19, no. 4, pp. 557-565.

[9] Ramírez-Treviño, A., Ruiz-Beltrán, E., Rivera-Rangel, I., López-Mellado, E., (2007) "On-line Fault Diagnosis of Discrete Event Systems. A Petri Net Based Approach", I*EEE Transactions on Automation Science and Engineering,* vol. 4, no. 1, pp. 31-39.

[10] Ruiz-Beltrán, E., Jiménez-Ochoa, I.,, Ramírez-Treviño. A., López-Mellado, E., Meda-Campaña, M., (2005), "Fault detection and location in DES modeled using Petri Nets," *in Proc. of the International Conference on Systems, Man and Cybernetics,* pp. 1645-1650.

Artículo recibido: 3 marzo de 2009

Aceptado para publicación: 29 mayo 2009