Distributed Fault Diagnosis Using Petri Net Reduced Models

J. Arámburo-Lizárraga, E. López-Mellado, A. Ramírez-Treviño
CINVESTAV Unidad Guadalajara.
Prol. López Mateos Sur 590,45090 Guadalajara, Jal. Mexico
e.lopez@gdl.cinvestav.mx

Abstract - This paper is concerned with the distributed diagnosis of Discrete Event Systems (DES) using reduced Interpreted Petri Nets (IPN) models. Two contributions are presented extending a previous work from the authors.
First, the condition of event detectability is relaxed over parts of the model where the faults are not expected; thus the diagnoser handles a reduced model. Second, a method for splitting the global model into communicating modules is proposed; this leads to the design of a set of distributed diagnosers.

Keywords: Reduced diagnosers, Distributed diagnosis, Discrete Event Systems, Petri nets.

1 Introduction

Opportune fault detection and isolation allows to minimise risks to humans and damages to equipment during the operation of large and complex systems; it constitutes an important first stage of a fault recovery process within an autonomous control system.

Fault diagnosis of automated discrete event systems (DES) received special attention in the literature through several approaches and methods. Regarding the model based approach several methods using finite automata (FA) [2], [3] and Petri Nets (PN) [4], [5], [6], [7], [1] have been proposed to address this problem. In these works it is studied the diagnosability property and fault detection schemes based on a centralised approach using the global model of the DES.

Distributed systems appeared in industry in some natural ways (computer and telecommunication networks, manufacturing process control and power systems, etc.) [15]. The challenges arising from the construction of distributed systems are the heterogeneity of its components, openness, which allow to add or replace components, security, scalability, failure handling, concurrency of components, and cleanness [16].

Recently, fault diagnosis of DES has been addressed through a distributed approach allowing to break down the complexity when dealing with large models [8], [9], [10], [11], [12], [13]. In [8] it is proposed a decentralised and modular approach to perform failure diagnosis based on Sampaio's results [2]. O. Contant et al. [10] and Y. Fencolé [9] propose incremental algorithms to perform diagnosability analysis based on [2] in a distributed way; they consider systems whose components evolve by the occurrence of events; the parallel composition leads to a complete system model intractable. In [11] S. Genc and S. Lafortune proposed a method that handles the reachability graph of the PN model in order to perform the analysis similar to [2]; based on design considerations the system model is partitioned into two labelled PN and it is demonstrated that the distributed diagnosis is equivalent to the centralised diagnosis.

In our approach we use the interpreted PN (IPN) model of the system including the possible faults that may occur. In a previous work we proposed a structural characterisation for diagnosability that avoid the reachability analysis and an on-line diagnoser scheme [6]. In this paper we extend that work by proposing a relaxation of the condition of event detectability over parts of the model where the faults are not expected; thus the diagnoser handles a reduced model. Also, a method for designing a set of distributed diagnosers by splitting the global model into communicating modules is proposed.

This paper is organised as follows. In section 2 basic definitions of PN and IPN are provided. Section 3 summarises the concepts and results for centralised diagnosis. Section 4 presents a methodology for designing reduced diagnosers. Finally, section 5 presents the methodology to split a global models into a set of communicating local models.

2 Background

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:
1. P = {p1, p2, ..., pm} and T = {t1, t2, ..., tn} are finite sets of vertices called respectively places and transitions,
2. I (O) : P × T → Z+ is a function representing the weighted arcs going from places to transitions (transitions to places), where 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 t ∈ I(p) denotes the set of all places p; such that I(p,t) ∈ 0 (O(p,t) = 0). Analogously, p ∈ O(t) (p,t) denotes the set of all transitions t such that O(p,t) ∈ 0 (I(p,t) = 0).
The pre-incidence matrix of $G$ is $C^- = [e_{ij}^-]$ where $e_{ij}^- = I(p_i, t_j)$; the post-incidence matrix of $G$ is $C^+ = [e_{ij}^+]$ where $e_{ij}^+ = O(p_i, t_j)$; the incidence matrix of this 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 $t_i$ is enabled at marking $M_i$ if $\forall p \in P, M_i(p) \geq I(p, t_i)$; an enabled transition $t_i$ can be fired reaching a new marking $M_{i+1}$ which can be computed as $M_{i+1} = M_i + C_{t_i}$, where $v_i(t_i) = 0$, and $v_i(t_i) = 1$, 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) an extension to PN that allow to associate input and output signals to PN models.

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

- $G$ is a PN structure.
- $\Sigma = \{a_1, a_2, \ldots, a_r\}$ is the input alphabet of the net, where $a_i$ is an input symbol.
- $\lambda: T \rightarrow \Sigma \times \{\varepsilon\}$ is a labelling function of transitions with the following constraint: $\forall t, t_i \in T, \lambda(t) = (\varepsilon, \varepsilon)$, if $\forall p, I(p, t_i) = \varepsilon$ and both $\lambda(t) = \varepsilon$, $\lambda(t_i) = \varepsilon$, then $\lambda(t) = \lambda(t_i)$. In this case the transition is called an internal system event.
- $\phi: R(Q, M_0) \rightarrow (Z^+)^\infty$ is an output function, that associates to each marking in $R(Q, M_0)$ an output vector. Here $q$ is the number of outputs.

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

A transition $t_i \in T$ of an IPN is enabled at marking $M_i$ if $\forall p_i \in P, M_i(p_i) \geq I(p_i, t_i)$. If $M_i(t) = \varepsilon$ and $\phi$ is present, then $t_i$ is enabled, hence $t_i$ must fire. If $M_i(t) = \varepsilon$ and $\phi$ is enabled then $t_i$ can be fired. When an enabled transition $t_i$ is fired in a marking $M_i$, then a new marking $M_{i+1}$ is reached. This fact is represented as $M_i \xrightarrow{t_i} M_{i+1}$; $M_{i+1}$ can be computed using the state equation:

$$M_{i+1} = M_i + C_{t_i} \quad (1)$$

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

**Definition 4:** A firing transition sequence of an IPN $(Q, M_0)$ is a transition sequence $o = t_1 ; t_2 \ldots$ such that $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots M_n \xrightarrow{t_n} \ldots$. The set $E(Q, M_0)$ of all firing transition sequences is called the firing language $E(Q, M_0) = \{ o = t_1 ; t_2 \ldots : M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \ldots M_n \xrightarrow{t_n} \ldots \}$.

**Definition 5:** Let $u = t_1 ; t_2 \ldots$ be a firing transition sequence. The Parikh vector $\sigma: T \rightarrow (Z^+)\infty$ of $o$ maps every $t \in T$ to the number of occurrences of $t$ in $u$.

According to functions $\lambda$ and $\phi$, transitions and places of an IPN $(Q, M_0)$ can be classified as follows.

**Definition 6:** If $\lambda(t_i) = \varepsilon$ the transition $t_i$ is said to be enabled. Otherwise it is non-enabled. A place $p_i \in P$ is said to be measurable if the $i$-th column vector of $\phi$ is non-null, i.e. $\phi(i, \cdot) \neq 0$. Otherwise it is nonmeasurable.

The following definitions relate the input and output symbol sequences with the firing transition sequences and the generated marking sequences. These concepts are useful in the study of the diagnosability property.

**Definition 7:** A sequence of input-output symbols of $(Q, M_0)$ is a sequence $o = (a_0, y_0)(a_1, y_1)\ldots(a_n, y_n)$, where $a_j \in \Sigma$ and $a_{ij}$ is the current input of the IPN when the output changes from $y_j$ to $y_{j+1}$. It is assumed that $a_0 = \varepsilon$, $y_0 = \phi(M_0)$ and $(a_{i+1}, y_{i+1})$ belongs to the sequence when:

- $(a_i, y_i)$ belongs to the sequence, $y_{i+1} \neq y_i$ and
- there exists no $y_{i+1}$ occurring after the occurrence of $y_i$ and before the occurrence of $y_{i+1}$.

**Definition 8:** Let $(Q, M_0)$ be an IPN. The set $\Lambda(Q, M_0) = \{ (o, \phi(o)) \}$ is a sequence of input-output symbols. The set of all input-output sequences of length greater or equal than $k$ is denoted by $\Lambda_k(Q, M_0)$, i.e.

$$\Lambda_k(Q, M_0) = \{ o = (a_0, y_0)(a_1, y_1)\ldots(a_n, y_n) \}$$

A transition sequence $o = (a_0, y_0)(a_1, y_1)\ldots(a_n, y_n)$ be a sequence of input-output symbols, then the firing transition sequence $s = \varepsilon Q(M_0)$ whose firing actually generates $o$ is denoted by $s$. The set of all possible firing transition sequences that could generate the word $o$ is defined as $Q(o) = \{ s \in \varepsilon Q(M_0) \wedge$ the firing of $o$ produces $o \}$.

**Definition 10:** The prefix of a sequence $s$ is another sequence $s'$ such that there exists a sequence $s''$ fulfilling that $s = s' \cdot s''$. The set of all prefixes of $s$ is denoted by $s$.

**Definition 11:** The set of all input-output sequences leading to an ending marking in the IPN (markings enabling no transition or only self-loop transitions) is denoted by $\Lambda_0(Q, M_0)$, i.e., $\Lambda_0(Q, M_0) = \{ o \in \Lambda(Q, M_0) \} \exists \sigma \in Q(o)$ such that $M_0 = \sigma \rightarrow M_1$ and $M_1$ enables no transition, or when $M_n \rightarrow \sigma \rightarrow \tau$ then $C(\sigma, \cdot) = 0$.

**Definition 12:** An IPN $(Q, M_0)$ is $(\Sigma, \Phi, \Lambda, D, \phi)$ described by the state equation (1) is event-detectable if 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 [14] gives a polynomial characteristic of event-detectable IPN.

**Lemma 13:** Let $(Q, M_0) = (\Sigma, \Phi, \Lambda, D, \phi)$ be an IPN described by the state equation (1). $(Q, M_0)$ is event-detectable if all $\phi$ columns are not null and different from each other.
3 Centralised Diagnosis

The main results on diagnosability and diagnoser design in a centralized approach presented in [6] are outlined below.

3.1 Diagnosability

The characterisation of input-output diagnosable IPN is based on the partition of \( R(Q,M_0) \) into normal and faulty markings: all the faulty markings must be distinguishable from other reachable markings representing normal functioning, \( R(Q,M_0) \) is composed of: a) the set of the faulty markings \( F = \{ M | \exists p \in P^f \text{ such that } M[p] > 0, M \in R(Q,M_0) \} \) and, b) the set of the normal states \( N = R(Q,M_0) - F \).

Definition 14: An IPN given by \((Q,M_0)\) is said to be input-output diagnosable in \( k < \infty \) steps if any marking \( M \in F \) is distinguishable from any other \( M_i \in R(Q,M_0) \) using any word \( w \in N^*(Q,M) \cup N^*(Q,M_i) \).

3.2 System modeling

Definition 15: The sets of nodes are partitioned into faulty nodes (\( P^f \), places coding faulty states, and \( T^f \), transitions leading to faulty states) and normal functioning nodes (\( P^n \) and \( T^n \)), so \( P = P^f \cup P^n \) and \( T = T^f \cup T^n \). \( p^N_i \) denotes a place in \( P^n \) of \((Q,M_0^n)\). Since \( P^N \subseteq P \) then \( p^N_i \) also belongs to \((Q,M_0)\). The set of risky places of \((Q,M_0)\) is \( P^R = P^N \cap T^f \). The post-nsk transition set of \((Q,M_0)\) is \( T^R = P^R \cap T^n \).

Figure 1 presents an IPN model of a system obtained with the modeling methodology presented in [S]. Notice that this model has three faulty states, represented by places \( p_5, p_{10}, p_{15} \). Function \( \lambda \) is defined as \( \lambda(t_1) = a, \lambda(t_2) = b, \lambda(t_3) = c \) for \( t_1 \), \( \lambda(t_2) = g, \lambda(t_3) = b, \lambda(t_4) = m \) for \( t_2 \), \( \lambda(t_5) = e \). Measurable places are \( p_6, p_7, p_8, p_{12}, p_{13}, p_{14}, p_{16}, p_9, p_{17}, p_{18}, p_{20}, p_2, p_3, p_4, p_5 \).\( T^f \) is \{t_1, t_2, t_3, t_4\}.

Figure 1. Global system model

3.3 Characterisation of diagnosability

The embedded normal behavior IPN \((Q^n,M^n_0)\) is the IPN included in \((Q,M_0)\) when \( P^f \) and \( T^f = \tau^f \) are not considered. In \((Q^n,M^n_0)\), the set of places is \( P^n = P - P^f \) and the set of transitions is \( T^n = T - T^f \).

Now, the following result provides sufficient structural conditions for determining the input-output diagnosability of an IPN model.

Theorem 16: Let \((Q,M_0)\) be a binary IPN obtained with the proposed modeling methodology, such that \((Q^n,M^n_0)\) is live, strongly connected and event detectable. Let \( \{X_1,...,X_t\} \) be the set of all T-semiflows of \((Q,M_0)\). If \( \forall p^N_i \in P^N, p^N_i \cap T^f \neq \emptyset \) the following conditions hold

1. \( \forall r, \exists \exists t, (t.i) \geq 1 \), where \( t.i \in (p^N_i) \cap T^f \).
2. \( \forall t_i \in (p^N_i) \cap T^f, t_i(f) = \{ p^N_i \} \) and \( \lambda(t) \in E \),
   then the IPN \((Q,M_0)\) is input-output diagnosable.

Proof: Included in [6]

3.4 Diagnosability test

Determining when an IPN is input-output diagnosable is reduced to the following tests[6]:

1. Binarity of \((Q,M_0)\). It is fulfilled because permissive and synchronous compositions of binary modules lead to binary IPN.
2. Liveness of modules can be tested efficiently; however the property is not preserved in the composed IPN model when it result from the arbitrary application of the composition operators. Thus some constraints in the application of synchronous and permissive compositions could guarantee that liveness is preserved. Such constraints may follow the rules introduced by Koh and DiCesare in [17] for module composition.
3. Event detectability and strongly connectedness are determined in polynomial time as well as detecting places \( p^N_i \leq P \) such that \( \{ p^N_i \} \cap T^f \neq \emptyset \). Then condition 2 of previous theorem is efficiently tested.
4. Finally, condition 1 can be verified in polynomial time. In this case we need to check that there exists not T-semiflow that does not include transitions in \( (p^N_i) \), \( p^N_i \in P^N \) such that \( \{ p^N_i \} \cap T^f \neq \emptyset \). Thus we need to check that the following linear programming problem have no solutions.

\[ X \geq 0 \]
\[ CX = 0 \]
\[ \forall t_i (p^N_i) - T^f \]

3.5 A centralised scheme for diagnosis

Diagnosability tests given above, provide the basis for designing an on-line diagnoser (see figure 2). Such diagnoser can detect a fault and locate faulty markings reached by an IPN. The proposed scheme for diagnosis [5][6] handles \((Q^n,M^n_0)\) which must evolve similarly to the system; the outputs of both the system and the model are compared, and, when there is a difference, a procedure is started computing the faulty marking.
4 Reduced Diagnosers

The condition of event detectability may be relaxed on \( t_i \in P^R \) and \( t_i \notin P^R \); these transitions are not involved during the detection of faults when \( (Q^n, M^n) \) is handled. We profit of this fact to obtain a reduced model \( (Q^{RM}, M^{RM}) \) containing the pertinent parts of \( (Q^n, M^n) \) regarding the modelled faults in \((Q,M_0)\). Consequently, the diagnoser monitors the system only when faults may arise.

The procedure performing this task is presented below.

**Algorithm 17: Reduced model**

Input: \((Q,M_0)\)  
Output: \((Q^{RM}, M^{RM})\)

- \( \forall p_i \in P^R \) then \( (Q^{RM}, M^{RM}_0) \) is defined as the subnet induced by:
  - \( P^{RM} = \{ p_i \} \cup p_i^- \)
  - \( T^{RM} = \{ p_i \in P^R \mid t \text{ is a transition that it fire when } p_i^- \text{ is marked} \}
  - Include the input symbols associated with \( T^{RM} \)
  - Include the output symbols associated with \( P^{RM} \)
  - If \( M_0(p_i) > 0 \) then \( M^{RM}_0(p_i) > 0 \)

Using the previous procedure on the model of figure 1, the obtained reduced model is shown in figure 3. Notice that in this example the number of places is reduced to a third of the complete.

![Figure 3. Diagnoser's reduced model](image)

Figure 3. Diagnoser's reduced model

The diagnoser scheme of figure 2 handles the reduced model and compares the outputs when this model is "active" i.e. when the first event detectable transition is fired. When a difference between the outputs is detected then the diagnosis procedure operates as described above.

5 Building a Distributed Diagnoser

The distributed scheme for on-line diagnosis is depicted in figure 4; it is a straightforward extension of the centraised scheme of figure 2. In this scheme there exist r "local" diagnosers corresponding to the module in which the system model is partitioned. Every diagnoser is composed of the embedded reduced normal behavior model corresponding to a module, and a diagnosis procedure; it handles subset of inputs and outputs regarding the module behavior. Since the local models may handle common events the diagnosers communicate for implementing synchronisation and inform of eventual faults arise within the corresponding part of a diagnoser module. The functioning of the diagnosers is the same but the precise behavior is determined by the corresponding local model.

![Figure 4. Distributed scheme for diagnosis](image)

5.1 Local IPN models

In order to build a decentralised diagnoser, the IPN model of the system must be partitioned or distributed. A module or partition of \((Q,M_0)\) is a subnet of \((Q,M_0)\), where different modules can share common nodes, but there is no arcs between modules. Thus two cases arises, when modules share transitions or places. Then two net transformation operations are used to transform the modules.

- When there exists a synchronisation transition \( t_i \) and we require to partition this transition to obtain two modules, then the transition \( t_i \) is duplicated in \( t_i \) and \( t_i^- \). Then places \( p_k \in t_i^- \) and \( p_k \in t_i^+ \) are also duplicated in \( p_i^- \) and \( p_i^+ \). All these duplicates nodes are connected in the same way that the original nodes. See figure 5.
- When there exists a conflict place \( p_i \) and we require to partition this place to obtain two modules, then the place \( p_i \) is duplicated in \( p_i^- \) and \( p_i^+ \); these duplicates places are connected in the same way that the original place. See figure 6.
One set of duplicated places is left in one module and the other set is left to the other module. Incoming arcs to places are dashed if they are coming from the other module. These dashed arcs indicate the communication among modules. Notice that the new places are implicit ones, thus places are dashed if they are output.

The following algorithm is proposed to get a model equivalent to the centralised one.

Algorithm 18: Partitioning IPN
Input: IPN \( N = (Q,M) \)
Output: IPN \( N' = (Q',M',\overline{E}) \) such that \( \overline{E}(Q',M') = \overline{E}(Q,M) \) and \( N(Q,M') = N(Q,M) \)

- \( \forall t_i \in T \) (synchronisation transitions) or \( \forall p_i \in P \) (conflict places) do
  - Duplicate \( t_i \) in \( T \) and \( t'_i \); places \( p_i \) and \( p'_i \) are also duplicated in \( P \) or
  - Duplicate \( p_i \) in \( P \) and \( p'_i \).

Build the incoming arcs, i.e., duplicates places (transitions) are connected in the same way that the original place (transition).

\( t_i (p_j) \) represents the chosen synchronisation transition (chosen conflict place) for partitioning the IPN.

The next step is to define the diagnoser modules and the interaction links that may be implemented in communication channels.

Algorithm 19: Building diagnoser modules.
Input: IPN \( N = (Q,M) \)
Output: \( D_h, i = 1,...,r \) (number of modules). The distributed diagnoser:

- Let \( N_M = \{ N'_1, ..., N'_r \} \) be the set of modules of net \( N \);
- Build the set of embedded normal behavior models \( N^N = \{ (Q^N_i, M^N_i) \} \);
- For each \( (Q^N_i, M^N_i) \), build the reduced model \( (Q^R_i, M^R_i) \) using the algorithm 17.
- Define \( k \) communication channels \( CC^i \) of module \( N'_i \), \( i = 1,...,k \), one for each outgoing dashed arc.

In the resulting distributed scheme the firing of each transition can be detected, since \( N^N = (Q^N_M, M^N) \) is event detectable on \( P^N \) and \( P^R \). Thus, every time that the firing of one transition is detected and this transition has an outgoing dashed arc, the information of this firing is propagated by the appropriate communication channels to the modules needing this information.

5.2 Local diagnoser procedure
Diagnosability tests given above (see subsection 3.4) provide the basis for designing a diagnoser. Such diagnoser can determine faulty markings reached by an IPN. The diagnoser scheme herein used (see figure 4) contains a copy of the underlying normal reduced behavior of the IPN model, a communication procedure that has the function to send the common events for implementing synchronisation and inform of eventual faults arising within the corresponding part of a diagnoser module and an algorithm to compute which faulty marking was reached.

The underlying normal reduced behavior will be named a local diagnoser-model IPN and it is defined as follows.

Definition 20: Let \( N^N = (Q^N, M^N) \) an IPN modeling a system, where the condition of theorem 16 hold; let \( N^R = (Q^R, M^R) \) be any IPN module of \( N \), where the net \( (Q^R, M^R) \) is the embedded normal reduced behavior of \( (Q, M) \) and the firing rule is defined as follows;

- If \( t_i \) was fired in the system and \( t_i \in T^R \), then it is fired in \( (Q, M) \).

There exists one diagnoser-model for each module.

5.2.1 Fault detection

The error between the system output and the local diagnoser-model output is \( E_t = \phi(M_t) \cdot \phi(M^R) \) where \( \phi(M_t) \) is restricted to the corresponding measurable places in \( P^R \). The following algorithm, devoted to detect which faulty marking was reached in \( N^R = (Q^R, M^R) \), is executed when \( E_{t_{10}} = 0 \).
Algorithm 21: Detecting Faulty Places

Inputs: $\phi(M, R^M)$ - the output of the local diagnoser-model.
\( \lambda(t_i) \) - the input symbol of the system
\( E_k \) - diagnoser error

Outputs: $p$ - the isolated faulty place

1. Constants: $\phi(C^N)$ - embedded normal behavior
2. Repeat
   2.a. Read $\phi(M^R)$
   2.b. If $\phi(M^R) = \phi(M_i^R)$ then computes
       \[ q = \phi(M^R) - \phi(M_i^R) \] (a column of $\phi(C^N)$)
   2.c. $i = \text{index of the column of } \phi(C^N) \text{ such that }$
       \[ \phi(C^N)(i, j) = q, \text{ then } t_i \text{ was fired}; p \leftarrow t_i \]
   2.d. If $E_k \neq 0$
       - $\forall p \in \{ t_i \} M^R(p) = 0$
       - $\forall p \in \{ t_i \} \{ p \} = 1$
       - Return ($p^*$)

Since $\left( Q^R, M^R_0 \right)$ is event detectable in $p^*$ and $p^*$, then step 2.b. will compute just one column index, moreover, and step 2.c. will compute just one place. Now, a diagnoser can be built as follows.

5.2.2 Fault location

Definition 22: Let $N = (Q, M_0)$ be an input-output diagnosable IPN. The triple $(N_0, E_k, A)$, where
- $N^R = (Q, M_0, A)$ is the distributed diagnoser model of $(Q, M_0)$
- $E_k$ is the error between the system output and the diagnoser-model output.
- $A$ is the algorithm Detecting Faulty Places

is named an input-output diagnoser for $(Q, M_0)$.

Acknowledgements

This work was partially supported by CONACYT in the scope of the project U41968Y.

6 Conclusions

We presented a framework for dealing with the problem of fault diagnosis of large and complex discrete event systems using a distributed approach. We addressed the relaxation of some conditions for structural characterisation of diagnosability in both centralised and distributed approaches, and we proposed an algorithm for designing a diagnoser model which represents the minimal behavior necessary for tracking the system model. The proposed distributed scheme uses a set of communicating local diagnosers that allows to handle fault location procedures while other components of the system remain working. Furthermore simultaneous faults occurred in the system can be diagnosed.

References