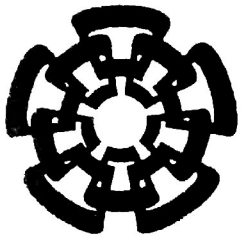

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Diagnóstico Distribuido Confiable de Sistemas de Eventos Discretos

Tesis que presenta:

Jesús Arámburo Lizárraga

para obtener el grado de:

Doctor en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Luis Ernesto López Mellado
Dr. Antonio Ramírez Treviño

Guadalajara, Jalisco, Junio de 2009.

Diagnóstico Distribuido Confiable de Sistemas de Eventos Discretos

Tesis de Doctorado en Ciencias
Ingeniería Eléctrica

Por:
Jesús Arámburo Lizárraga
Maestro en Ciencias con Especialidad en Ingeniería Eléctrica
Centro de Investigación y Estudios Avanzados del IPN
2003-2005

Becario de CONACYT, expediente no. 182724

Directores de Tesis
Dr. Luis Ernesto López Mellado
Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, Junio de 2009.

Reconocimiento

A mis asesores, Dr. Luis Ernesto López Mellado, Dr. Antonio Ramírez Treviño; por su gran apoyo para la dirección de esta tesis y sobre todo por los conocimientos que me transmitieron para poder realizar esta investigación.

Al Centro de Investigación y Estudios Avanzados del IPN (CINVESTAV) Unidad Guadalajara, por el apoyo recibido para realizar mis estudios de Maestría y Doctorado.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por otorgarme apoyo económico durante el tiempo de la investigación.

Agradecimientos

A mis padres, Raymundo Arámburo Lizárraga, Cleotilde Lizárraga Sánchez por el amor que me han otorgado y por inspirarme a crecer profesionalmente.

A mis hermanos Roberto, Raymundo, Delia y Griselda por el cariño y apoyo incondicional que me han dado y un agradecimiento muy especial a mi hermana Yolanda por ser una guía en mi camino ya que el amor y soporte que me ha brindado es el tesoro más valioso que puedo tener.

Sucedió una mañana en la primavera del 2002, al despertarme mis alas se habían roto, no tenía ganas de levantarme y transitar por la vida, quería seguir volando por las nubes, no ambicionaba caminar en la tierra. Asumí días de obscuridad, me creía sin alma, sin espíritu, aterrado en una profunda soledad, hasta que un día Dios apareció en mi vida, iluminó con su presencia mi habitación y me dijo: Angel mío el momento ha llegado, levántate y construye tu camino que yo estaré a tu lado.

Gracias Dios por la vida que hemos compartido.

Jesús Arámburo Lizárraga

Junio de 2009

Diagnóstico Distribuido Confiable de Sistemas de Eventos Discretos

Resumen

En este trabajo se estudia el problema de diagnóstico distribuido en Sistemas de Eventos Discretos (*SED*) modelados con redes de Petri interpretadas (*RPI*).

Una primera contribución de esta tesis consiste en una extensión a una caracterización previa de modelos diagnosticables permitiendo diagnosticar una clase mayor de modelos en *RPI*. La diagnosticabilidad de un *SED* distribuido se caracteriza en términos de la diagnosticabilidad del modelo global y se propone un esquema confiable de diagnosticadores distribuidos en base a los siguientes tipos de redundancia: duplicación y triple redundancia modular (*TRM*). Las técnicas de redundancia aplicadas al diagnosticador distribuido permiten detectar y localizar el funcionamiento incorrecto en el diagnosticador.

Otra contribución importante es un método para la distribución de un modelo el cual minimiza la comunicación entre los diagnosticadores distribuidos. Este método está basado en la representación de las dependencias de las faltas en el modelo global del sistema mediante un grafo de comunicación; este grafo se utiliza en un procedimiento el cual determina los sub-modelos para los diagnosticadores garantizando la mínima comunicación para su operación.

Reliable Distributed Diagnosis of Discrete Event Systems

Abstract

This work deals with distributed fault diagnosis of Discrete Event Systems (*DES*) modeled by interpreted Petri nets (*IPN*).

A first contribution of this thesis is an extension to a previous characterization of diagnosable models allowing diagnosing a broader class of *IPN* models. The diagnosability property of a distributed *DES* is characterized in terms of the diagnosability the global model; also it is proposed a scheme for reliable distributed diagnosers based on the following types of redundancy: duplication and triple modular redundancy. Redundancy techniques applied to distributed diagnosers can detect and localize a malfunction in the diagnoser.

Other important contribution is a method for the distribution of a global model, which minimizes the communication among the distributed diagnosers. This method is based on the representation of the system model faults dependencies through a communication graph; this graph is processed by an algorithm that determines the sub-models for the diagnosers, which guarantee minimal communication for their operation.

Índice General

Introducción

Los sistemas de producción desarrollados por el hombre son grandes y complejos por lo que requieren mecanismos oportunos de detección y localización de las faltas que pueden presentarse en los sistemas.

En términos generales una falta se considera un cambio inesperado en algún componente de un sistema, sin desviarse el comportamiento especificado del sistema. Las consecuencias de un diagnóstico de faltas oportuno permiten reducir los riesgos de lesiones en seres humanos, el impacto negativo en el medio ambiente y minimizar pérdidas económicas.

Los métodos de diagnóstico de faltas, básicamente, pueden clasificarse en dos tipos: aquellos métodos que no utilizan modelos matemáticos del sistema a diagnosticar y los métodos basados en el modelo. Los sistemas físicos usualmente suelen representarse por modelos matemáticos, donde las propiedades del modelo reflejan la naturaleza del sistema.

En el área de sistemas de eventos discretos (*SED*) dos formalismos son lo más utilizados para obtener el modelo: los autómatas finitos (*AF*) y las redes de Petri (*RP*). El modelo en *AF* o *RP* captura el comportamiento normal y de falta del sistema, con este modelo se puede verificar si es posible detectar la ocurrencia de las faltas del sistema en un tiempo finito; cuando se cumple esta propiedad se dice que el sistema es *diagnosticable*. Una entidad que monitorea la operación de un sistema y que detecta una falta y la localiza es llamada *diagnosticador*. Esta entidad generalmente apoya sus funciones en base al modelo del sistema.

La complejidad de las tareas del diagnosticador aumentará en proporción a la talla del sistema; de igual forma aumentará la posibilidad de mal funcionamiento. Esto ha motivado el estudio de la distribución del diagnosticador con el fin de mejorar la eficiencia y la confiabilidad de las entidades realizando esta función. Trabajos recientes se han enfocado sobre el diagnostico distribuido [Benveniste, et al., 2003], [Contant O., et al., 2004], [Debouk, et al. 2000], [Genc y Lafortune, 2003], [Jiroveanu y Boel, 2003], [Pencolé, 2004] y [Arámburo-Lizárraga, et al., 2005, 2007[a], 2007[b], 2008[a], 2008[b]].

Esta tesis trata sobre Diagnóstico Distribuido en Sistemas de Eventos Discretos, donde el modelo del sistema se expresa en redes de Petri Interpretadas (*RPI*), las cuales son una extensión de las redes de Petri. Las *RPI* permiten capturar el comportamiento medible y no medible del sistema así como eventos internos y manipulables que son parte de la entrada del sistema.

El objetivo planteado inicialmente en esta tesis ha sido la definición de una metodología para la concepción de diagnosticadores distribuidos confiables realizando su función de manera eficiente. Este objetivo ha sido alcanzado mediante las propuestas que tratan sobre la diagnosticabilidad distribuida y el diseño de una red confiable de diagnosticadores distribuidos. A grosso modo, el trabajo presenta dos contribuciones.

Una primera contribución de esta tesis consiste en una extensión a una caracterización previa de modelos diagnosticables permitiendo diagnosticar una clase mayor de modelos en *RPI*. La diagnosticabilidad de un *SED* distribuido se caracteriza en términos de la diagnosticabilidad del modelo global y se propone un esquema confiable de diagnosticadores distribuidos en base a los siguientes tipos de redundancia: duplicación y triple redundancia modular. Las técnicas de redundancia aplicadas al diagnosticador distribuido permiten detectar y localizar el funcionamiento incorrecto en el diagnosticador.

Otra contribución importante es un método para la distribución de un modelo el cual minimiza la comunicación entre los diagnosticadores distribuidos. Este método está basado en la representación de las dependencias de las faltas en el modelo global del sistema mediante un grafo de comunicación; este grafo se utiliza mediante un procedimiento el cual determina los sub-modelos para los diagnosticadores garantizando la mínima comunicación para su operación.

El contenido de esta tesis está organizado como sigue.

El capítulo 1 sintetiza los conceptos básicos sobre diagnóstico, y el trabajo relacionado. Ahí se realiza el planteamiento del problema en detalle.

En el capítulo 2 se presentan los conceptos básicos de redes de Petri y redes de Petri Interpretadas que son el formalismo utilizado para modelar *SED*. Se captura el comportamiento normal y de falta de los sistemas en el modelo obtenido. Se define la propiedad de diagnosticabilidad en modelos centralizados, dicha propiedad, se basa en las entradas, salidas y la estructura del modelo del sistema, finalmente se define el esquema de diagnóstico en línea utilizando la herramienta de diagnosticadores reducidos.

En el capítulo 3 se propone el esquema de diagnosticador distribuido para sistemas de gran tamaño y funcionamiento complejos. Se define la propiedad de diagnosticabilidad a un enfoque distribuido.

En el capítulo 4 se define el problema de optimización relacionado con la obtención de una distribución óptima, donde los modelos diagnosticadores preservan la propiedad de diagnosticabilidad. La distribución obtenida requiere un mínimo de comunicación entre los diagnosticadores distribuidos.

Finalmente, se presentan las conclusiones de este trabajo, así como algunos aspectos del trabajo que merecen un estudio adicional.

Capítulo 1 Diagnóstico de Faltas en Sistemas de Eventos Discretos

Resumen: *Este capítulo presenta los términos utilizados en el problema de Diagnóstico de Faltas en Sistemas de Eventos Discretos (SED). Se incorpora una revisión de los trabajos relacionados con Diagnóstico de Faltas, mostrando las ventajas y desventajas de las técnicas propuestas. Posteriormente, se define el problema que se investiga en este trabajo, los objetivos y contribuciones.*

1.1. Conceptos Relacionados con Diagnóstico de Faltas

Los términos falta, falla y error suelen usarse indistintamente para denotar un mal comportamiento en el sistema, sin embargo, existen diferencias entre éstos. A continuación se definen los términos falta, falla y error [Jalote, 1994].

Definición 1.1. Una falla se presenta en el sistema cuando el comportamiento del mismo se desvía del comportamiento requerido.

Definición 1.2. Un error es la diferencia observable entre lo que está especificado y lo que realmente ocurre en el sistema.

Definición 1.3. Una falta es la fuente original de cualquier problema que puede conducir a un error en el comportamiento especificado por el sistema.

En este trabajo se utiliza el término de falta, para especificar que algún elemento o componente del sistema ha fallado pero dicha falla no se refleja a nivel sistema. Las faltas se clasifican de acuerdo al tiempo que permanecen en el sistema. La clasificación propuesta en [Jalote, 1994] define dos tipos de faltas: transitorias y permanentes.

Definición 1.4. Las faltas transitorias son faltas de duración limitada, causadas por comportamientos erróneos del sistema, o bien, debido a interferencias externas.

Definición 1.5. Las faltas permanentes son aquellas que una vez que los componentes fallan, éstos nunca (o por lo menos por un periodo largo de tiempo) trabajan correctamente de nuevo.

Un sistema debe de tener un conjunto de propiedades que aseguren un desempeño óptimo del mismo. A continuación se definen algunas de ellas.

Definición 1.6. *Confiabilidad.* Propiedad del sistema para desempeñar una tarea específica bajo condiciones establecidas, durante un periodo de tiempo dado [Chen y Patton, 1999].

Definición 1.7. *Seguridad.* Propiedad del sistema de no causar daños a personas, equipo o al ambiente [Chen y Patton, 1999].

Definición 1.8. *Disponibilidad.* Propiedad del sistema para operar satisfactoriamente y efectivamente en cualquier instante de tiempo [Chen y Patton, 1999].

Definición 1.9. *Tolerante a faltas.* Un sistema se considera tolerante a faltas cuando puede enmascarar la presencia de faltas en el sistema usando redundancia y continúa su operación normalmente o con cierto grado de degradación, o en el peor de los casos se presenta un paro temporal en su ejecución dejando al sistema en un estado seguro. Estos sistemas pueden experimentar diferentes niveles de tolerancia como:

1. *Tolerancia a faltas completa*, cuando el sistema continúa funcionando en presencia de faltas, aunque sea por un período de tiempo limitado sin tener una pérdida significante de funcionalidad y rendimiento.
2. *Degradación suave*, cuando el sistema continúa funcionando en presencia de errores, aceptando una degradación parcial del funcionamiento y rendimiento durante la etapa de recuperación.
3. *Detención segura*, el sistema se detiene en un estado que mantiene su integridad mientras su operación está en paro temporal.

La tolerancia a faltas en sistemas involucra cuatro fases [Jalote, 1994]:

- Detección del error,
- Confinamiento del daño,
- Recuperación del error
- Tratamiento de la falta y continuar con el servicio del sistema.

La *detección del error* es la fase que permite detectar la presencia de una falta en el sistema, cuando el error es detectado se dice que existe una falla en algún subsistema (componente) del sistema. *Confinamiento del daño* implica localizar al componente que falla e identificar y delimitar cualquier daño causado por esta falta del sistema. La fase de *recuperación del error* significa dirigir al sistema del estado corrupto a otro estado a partir del cual se continúe con una operación normal o con un nivel de tolerancia especificado previamente, evitando que la falla del componente propague su efecto a todo el sistema. La última fase *tratamiento de la falta y continuar con el servicio del sistema* se aplica al identificar al componente que tiene la falta, el sistema se repara de tal manera que el componente con falla no se use o sea usado en una configuración distinta. Finalmente, una vez que el componente se repara, el servicio normal puede continuar.

La redundancia es una forma para soportar la tolerancia a faltas en sistemas, y es definida en [Jalote, 1994] como:

Definición 1.10. Redundancia implica utilizar componentes adicionales para detectar las faltas del sistema, recuperando el comportamiento deseado de los componentes que fallan. Son componentes extras que no son necesarios para el funcionamiento correcto del sistema, esto es, un sistema trabaja correctamente sin redundancia, si ninguna falla se presentara en los componentes. Redundancia puede ser aplicada en hardware, software o tiempo.

En este trabajo se aborda el problema de diagnóstico de faltas basado en modelos, esto es, se consideran solamente las dos primeras fases de un sistema tolerante a faltas (detección del error y confinamiento del daño) y se supone que se conoce un modelo del comportamiento normal y otro con falla del sistema. A continuación se definen los conceptos de sistema de diagnóstico de faltas [Palade, 2006] y diagnóstico basado en modelos [Chen y Patton, 1999].

Definición 1.11. Un sistema de diagnóstico de faltas es un sistema de monitoreo que es usado para detectar y localizar las faltas así como para el confinamiento del daño en el sistema. El sistema realiza las siguientes tareas:

- Detección de la falta – para indicar si una falta ocurrió o no en los componentes del sistema.
- Aislamiento de la falta – para determinar la localización de la falta.
- Identificación de la falta – para estimar el tamaño y naturaleza de la falta.

Definición 1.12. El diagnóstico de faltas basado en el modelo es definido como la determinación de las faltas en un sistema comparando las medidas disponibles de un sistema con información a priori representada por un modelo matemático/analítico del sistema, a través de la generación de cantidades residuales y su análisis.

El diagnóstico de faltas basado en modelos se interpreta como la detección, aislamiento y caracterización de las faltas del sistema utilizando las medidas disponibles del sistema, con información a priori representada por algún modelo matemático del sistema.

En los últimos años se ha mostrado gran interés por estudiar el problema de diagnóstico utilizando arquitecturas descentralizadas o distribuidas [Da Silveira et al., 2002], [Genc y Lafortune, 2005], [Jiroveanu y Boel, 2005], [Provan, 2000], [Su, 2002], entre otros autores. Los sistemas manejan grandes cantidades de información, por lo que, se utilizan arquitecturas descentralizadas, distribuidas o modulares que manipulan la información.

A continuación se define un sistema distribuido.

Definición 1.13. Un sistema distribuido consiste de un conjunto de computadoras separadas geográficamente, pero conectadas entre sí por medio de una red de comunicaciones. Las computadoras funcionan de manera concurrente para desempeñar alguna tarea.

En este trabajo se realiza diagnóstico de *SED* modelado con *RPI* utilizando diagnosticadores centralizados y distribuidos con o sin redundancia asegurando diagnóstico del sistema a pesar de que existan fallas en los diagnosticadores, el esquema distribuido propuesto es confiable para resolver el problema de diagnóstico de faltas basado en modelos. A continuación, se resumen algunos trabajos previamente publicados sobre diagnóstico de faltas mediante modelos de eventos discretos. Estos trabajos utilizan los formalismos de autómatas y redes de Petri para obtener el modelo del sistema. Se presenta un análisis de las ventajas y desventajas de cada uno de ellos.

1.2. Monitoreo y diagnóstico de faltas basado en modelos de Eventos Discretos

Los Sistemas de Eventos Discretos (*SED*) son sistemas dinámicos, cuyo espacio de estados es numerable, aunque posiblemente infinito, y donde el estado cambia repentinamente en respuesta a eventos que suceden, en general, de forma asíncrona [Silva, 1985]. Los protocolos de

comunicación, sistemas de manufactura, sistemas de tráfico urbano, entre muchos otros, son ejemplo de este tipo de sistemas, por lo tanto son de gran importancia. Existen diferentes investigaciones que se enfocan al modelado, análisis, control y diagnóstico de faltas de *SED*, donde el trabajo de esta tesis se enfoca al diagnóstico de faltas utilizando diagnosticadores centralizados y distribuidos. A continuación se define formalmente un *SED*.

Definición 1.14. Un Sistema de Eventos Discretos (*SED*) es un sistema de estado discreto y dirigido por eventos, esto es, su evolución depende enteramente de la ocurrencia de eventos discretos asíncronos sobre el tiempo [Cassandras y Lafortune, 2008].

A continuación, se presentan algunas investigaciones sobre el diagnóstico de faltas basados en el modelo de eventos discretos. Los trabajos consideran los formalismos de autómatas y de redes de Petri utilizando un ambiente centralizado y distribuido en el modelo del sistema.

1.2.1. Diagnóstico Centralizado de Faltas en SED

La propiedad de diagnosticabilidad y esquemas de detección de faltas han sido ampliamente estudiados utilizando estrategias centralizadas, las cuales usan modelos globales de Sistemas de Eventos Discretos. Diagnosticabilidad es la propiedad que se estudia en un modelo de un sistema para saber si es posible detectar y localizar los estados de falta en un número finito de pasos.

En trabajos como [Sampath, et al., 1995] y [Sampath, et al., 1996] se propone un método para modelar un *SED* usando un autómata finito, se introduce la noción de diagnosticabilidad y proponen una técnica para la construcción del diagnosticador. En estos trabajos el modelo del sistema incluye las posibles faltas que son representadas por medio de transiciones incontrolables o internas llamadas transiciones-ε. El modelo sin transiciones-ε se transforma en un nuevo modelo llamado diagnosticador, el cual sirve para verificar la propiedad de diagnosticabilidad. Sin embargo, la propuesta de construcción de diagnosticador y la prueba de diagnosticabilidad tienen complejidad exponencial en el número de estados del sistema.

Se han realizado extensiones a los trabajos propuestos por Sampath. En [Hashtrudi, et al., 2003] se propuso un método que permitía al diagnosticador y al sistema iniciar su ejecución en condiciones iniciales diferentes. Sin embargo, al no ser inicializados en el mismo instante de tiempo, se genera un espacio de estados mayor que en el propuesto en Sampath.

En [Jiang, et al., 2001] se propuso un algoritmo polinomial para la prueba de diagnosticabilidad sin la necesidad de construcción del diagnosticador. El algoritmo se basa en los resultados de Sampath.

En [Chung, et al., 2003] se extendieron los resultados de Sampath al formalismo de redes de Petri (*RP*); se presentó la construcción de un diagnosticador para sistemas modelados con *RP*, sin embargo, se utiliza el grafo de alcanzabilidad de la *RP* para analizar la propiedad de diagnosticabildad. [Giua y Seatzu, 2005] calculan el estado de falta de sistema utilizando un algoritmo en línea que estima los marcados de la *RP* a partir del disparo de eventos observables, pero no realizan la prueba de diagnosticabilidad.

Recientemente, el diagnóstico de faltas de *SED* se ha abordado a través de una estrategia distribuida permitiendo disminuir la complejidad cuando se trabajan con sistemas grandes y complejos.

1.2.2. Diagnóstico Distribuido de Faltas en SED

En [Da Silveira et al., 2002] se propone un método de descomposición de redes de Petri utilizando la teoría de p-invariantes. A partir de un modelo de referencia global se obtiene una representación modular del comportamiento del sistema, donde los modelos obtenidos cuentan con redundancia parcial. Cada submodelo describe el comportamiento de un recurso (ó un conjunto de ellos) y sus interacciones con otros recursos. La redundancia de los submodelos significa la relación de acceso a recursos del sistema por varios submodelos. Se propone un procedimiento de comunicación el cual se inicia durante la ejecución de una actividad, cuando la actividad depende de varios recursos, entonces los módulos que controlan esos recursos tienen que comunicarse con el propósito de actualizar su información y sincronizar sus acciones. Las principales limitaciones de este trabajo son: la metodología de particionado debe asegurar que todos los recursos deben tener sus actividades representadas en al menos un proceso del submodelo y la decisión de diagnóstico del sistema se realiza de manera centralizada.

En [Fabre et al., 2000] se enfocan a sistemas obtenidos por la composición paralela de varios subsistemas modelados con *AF*. Cada subsistema puede ser visto como un sistema de eventos discretos estocástico. El objetivo es recuperar una trayectoria global muy similar a la generada por un sistema centralizado a partir de las secuencias asíncronas de los dos subsistemas. Se utilizan algoritmos Viterbi basados en estados locales y la trayectoria global se construye recursivamente. La arquitectura se aplica a monitoreo de sistemas distribuido de redes de telecomunicaciones. Sin embargo, manejar estados globales del sistema en sistemas distribuidos y tratar de encontrar la trayectoria más adecuada bajo la forma de una secuencia de eventos requieren de una complejidad alta en tiempo por la combinatoria generada.

En [Genc y Lafortune, 2005] presentan un nuevo algoritmo distribuido para detección y localización de faltas en *SED* modelados con redes de Petri. Consideran el caso de sistemas modulares que consisten de un conjunto de redes de Petri etiquetadas con lugares comunes y transiciones diferentes, la motivación para etiquetar las transiciones que ponen o quitan marcas de lugares comunes con eventos observables permite la comunicación entre diagnosticadores. Proponen una extensión del algoritmo llamado DDC-2 presentado en [Genc y Lafortune, 2003], denominando a este nuevo algoritmo DDC-M, utilizando comunicación para m módulos. El algoritmo se presenta en 2 partes: a) Parte 1. Se realiza la actualización del estado del diagnosticador local de acuerdo al monitoreo de los eventos del sistema, y determina si es necesaria la generación de mensajes de comunicación sobre la ocurrencia de un evento observable común en un módulo, b) Parte 2. Corresponde a la actualización del estado del diagnosticador sobre la recepción de un mensaje de otro módulo. Definen una operación llamada Merge que combina los estados locales de los diagnosticadores para obtener el estado global equivalente y de esta manera comprueba que el algoritmo DDC-M es correcto en el sentido de que reconstruye un estado equivalente a un estado generado por un diagnosticador centralizado. Uno de los problemas que tiene este trabajo es la comunicación utilizada por los diagnosticadores, los mensajes van creciendo, ya que adjuntan al mensaje información adicional para garantizar una transmisión confiable y por lo tanto trae problemas de retraso en la comunicación.

En [Haar et al., 2005] abordan el problema de diagnóstico en sistemas de información ó de telecomunicaciones mediante redes de Petri. El sistema es monitoreado a través de sitios, donde cada sitio guarda o colecciona las alarmas (o eventos) generados por el sistema, posteriormente se procesa la información por un supervisor central. Cada sitio local tiene una vista parcial del sistema, y su tiempo local no está sincronizado con el resto de los sitios. El tiempo se regula en cada nodo. Usan el formalismo de desdoblamiento (unfoldings) de *RP* para representar las

ejecuciones posibles de una secuencia de señales emitidas por un módulo. Además utilizan los grafos gramaticales como sistemas de eventos discretos dinámicos. Los grafos son generados por un estado inicial y una colección de reglas que van reconstruyendo el estado actual del sistema. Se obtienen todas las posibles secuencias de transiciones que pueden explicar la secuencia de alarmas emitidas por el sistema, se consideran las relaciones entre módulos y las relaciones cuando se crean nuevos grafos de acuerdo a las reglas de reconfiguración, lo cual no resulta en tiempo computacional eficiente.

En [Jiroveanu y Boel, 2005] modelan al sistema de una planta como varios subsistemas modelados con *RP* que interactúan entre sí. Presentan un agente diagnosticador (d-agente) el cual tiene el modelo local *RP*, recibe las observaciones locales y puede intercambiar información limitada con agentes vecinos. Las interacciones entre diferentes subsistemas se representan por marcas que pasan vía lugares (no observables) comunes desde un subsistema a otro, sin embargo, en cada modelo de *RP* de un subsistema existe al menos un evento observable (transición) en cada camino desde un lugar de entrada a un lugar de salida. Diseñan un protocolo de comunicación para intercambiar información entre los diferentes d-agentes del modelo, se valida que el diagnóstico distribuido es equivalente al realizado por un diagnosticador centralizado. Utilizan un reloj global para coordinar todo el proceso de comunicación de d-agentes. Un reloj global para la sincronización de mensajes entre los diferentes diagnosticadores es algo restrictivo, en sistemas puramente distribuidos no hay un reloj global que controle los estados locales.

En [Provan, 2000] presentan una arquitectura de control y diagnóstico distribuido para sistemas distribuidos embebidos. Proponen un marco de trabajo de diagnóstico distribuido basado en redes causales, y prueban la diagnosticabilidad del sistema basada en la diagnosticabilidad de componentes. Algunos subsistemas no son diagnosticables con la información local de los sensores y requieren información de otros módulos. Obtienen el modelo de la planta en términos de redes causales y se basa en un conjunto de variables y una instancia de esas variables. Hay variables representando las fallas de los componentes, las cuales son variables no observables, y para propósitos de control particionan las observables en sensores y actuadores. Presentan una función de descomposición para separar un modelo centralizado en una colección de bloques (módulos) interconectados. Definen bloque diagnosticable (como diagnosticabilidad por módulos). Además definen dos tipos de diagnosticabilidad: fuerte y débil. La fuerte es que los bloques con la información local son capaces de detectar las faltas, la débil necesita información de otros bloques. Proponen un lema donde la propiedad de diagnosticabilidad distribuida no es más fuerte que la centralizada, es decir, no existe un conjunto de faltas que sea diagnosticable de manera distribuida mientras que centralizada no. Mencionan tres formas de garantizar la diagnosticabilidad de manera distribuida: 1) el intercambio de mensajes entre bloques, 2) agregar ecuaciones extras a bloques particulares junto con intercambio de mensajes, 3) agregar sensores extras a cada bloque (con esta estrategia se logra fuerte diagnosticabilidad, pero aumentan los costos). Una de las desventajas de la estrategia de envío de mensajes es que si el mensaje es extenso, dirige a tener un tráfico de información. Para la estrategia de agregar ecuaciones, comentan que éstas se obtienen de la interconexión de la colección de bloques del sistema, son dependientes del modelo. Las desventajas de esta estrategia son: a) en el caso peor la complejidad de diagnóstico de un bloque equivale a la de un modelo centralizado y b) es difícil definir las ecuaciones locales de manera independiente al modelo, no existe una metodología general para lograrlo.

En [Su, 2002] proponen una arquitectura basada en autómatas para construir un diagnosticador. Utilizan un método de diagnóstico distribuido basado en computación local y comunicación. El método propuesto es altamente escalable y robusto a faltas parciales de todo el diagnosticador. Cada componente local tiene su propio diagnosticador, el cual se construye de acuerdo al conocimiento del componente que forma parte. Cada diagnosticador local se conecta

con los otros diagnosticadores de acuerdo a las relaciones de entrada-salida asociadas a los componentes locales. El diagnóstico local en cada diagnosticador se basa principalmente en la observación local y la comunicación sólo se usa para propósitos de refinamiento. De esta manera si algún diagnosticador local ó canal de comunicación falla, otros diagnosticadores locales podrían refinar entre ellos de acuerdo a los canales de comunicación no dañados. Los procedimientos y definiciones están dados en base a un ciclo de trabajo (la duración de cada instancia que tiene una función predefinida. El ciclo de trabajo se divide en *m* subintervalos de tiempo. Hay un reloj global disponible para cada diagnosticador local para sincronizar la comunicación. El algoritmo para obtener el estimado global es exponencial y el reloj global no es útil en sistemas asíncronos. Así como el algoritmo de todos los estimados globales es NP-hard.

1.2.3. Diagnóstico de Faltas utilizando la estructura del SED

Para evitar el análisis de alcanzabilidad para determinar la propiedad de diagnosticabilidad, se han propuesto una serie de trabajos [Ramírez-Treviño, et al., 2004], [Ruiz-Beltrán, et al., 2004, 2005, 2006 y 2007] que se basan en la estructura del modelo que incorpora el comportamiento normal y de falta del sistema para evaluar la diagnosticabilidad de manera eficiente proponiendo algoritmos polinomiales para conocer cuando una *RPI* es diagnosticable entrada-salida con respecto a las faltas modeladas.

El esquema estructural para diagnóstico que se presenta en este trabajo es una extensión a los trabajos [Ruiz-Beltrán, et al., 2004, 2005, 2006 y 2007]. El objetivo es proveer algoritmos polinomiales para verificar la propiedad de diagnosticabilidad en *SED* que son parcialmente observables, evitando el análisis de alcanzabilidad de otras propuestas.

El modelo es un aspecto importante para el análisis estructural, ya que debe de incorporarse los estados de falta que son parte de los componentes del sistema. El análisis estructural de modelo determina la presencia de faltas en los componentes modelados en el *SED*, además permite distinguir si el sistema modelado es o no diagnosticable.

1.3. Definición del Problema

Este trabajo trata del diagnóstico de faltas en *SED* (ver figura 1.1) utilizando primero un diagnosticador centralizado para posteriormente adaptar los resultados a una arquitectura distribuida aprovechando la naturaleza de los sistemas grandes y complejos. Se puede utilizar la redundancia en los modelos distribuidos para manejar errores en los diagnosticadores.

1.3.1. Descripción General del Problema de Diagnóstico Distribuido Confiable en SED

El esquema de la figura 1.1 muestra el diagnóstico de faltas en el sistema, dicho esquema refleja que se utiliza el término falla para fallas en los subsistemas mientras que falta será aplicado al sistema. Existen diferentes estrategias para diagnosticar, este trabajo se enfoca en la construcción de un diagnosticador, presentando dos enofques: centralizado y distribuido. El diagnosticador distribuido es confiable ya que permite manejar dos tipos de modelos: reducidos y modelos redundantes con las técnicas de duplicación y Triple Redundancia Modular, permitiendo que el diagnosticador de algún subsistema pueda fallar. Además, se presenta el problema de obtener una

distribución del sistema global. La distribución está compuesta por módulos del sistema que serán parte del diagnosticador distribuido, donde existe comunicación mínima en el diagnosticador.

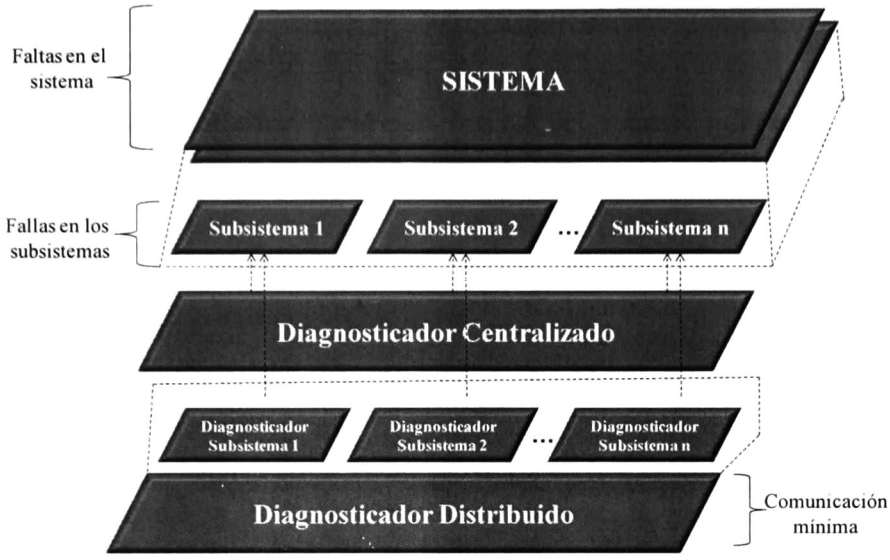

Fig. 1.1. Diagnóstico Distribuido de SED

Los problemas que se estudian en esta tesis son:

- Realizar una extensión a la caracterización de la propiedad de diagnosticabilidad centralizada de *SED* propuesta en [Ruiz-Beltrán, et al., 2007].
- Diseño de un diagnosticador distribuido para la detección y localización de las faltas en línea.
- Diseño de un diagnosticador distribuido que incorpore redundancia (duplicación y Triple Redundancia Modular) en los modelos del diagnosticador.
- Manejar fallas en el diagnosticador distribuido. Un diagnosticador puede fallar si no realiza la función de diagnóstico del sistema, o porque no envía los mensajes de comunicación.
- Obtener un diagnosticador distribuido óptimo a partir de un grafo de comunicación dependiente entre faltas del sistema modelado. Debe existir comunicación mínima en el diagnosticador distribuido.

1.3.2. Objetivos y contribuciones de la tesis

En esta tesis se aborda el problema de diagnóstico de faltas en modelos de *SED* utilizando un diagnosticador distribuido. Se utilizan las redes de Petri Interpretadas (*RPI*) para representar el comportamiento normal y de falta de los *SED*. Se incorpora la redundancia en los modelos del diagnosticador distribuido, para darle confiabilidad al esquema de diagnóstico en línea. Utilizando las consideraciones anteriores, se plantean los siguientes objetivos y contribuciones en esta tesis:

Objetivos generales

- Extender la clase de sistemas modelados con *RPI* que pueden ser diagnosticables propuesta en [Ruiz-Beltrán, 2007].

- Adaptar la propiedad de diagnosticabilidad a una arquitectura distribuida.

- Proponer un esquema de diagnóstico distribuido confiable utilizando diagnosticadores redundantes que puedan fallar.

- Obtener un diagnosticador distribuido óptimo del sistema utilizando el criterio de comunicación mínima entre modelos diagnosticadores.

Contribuciones de la tesis

Las contribuciones de esta tesis son las siguientes:

- Se puede diagnosticar una subclase mayor de *RPI* y se traslada la propiedad de diagnosticabilidad centralizada a una arquitectura distribuida.

- Se proponen tres diagnosticadores (centralizado, distribuido y distribuido redundante) eficientes y confiables ya que una vez que se garantiza la diagnosticabilidad entonces el diagnosticador detecta y localiza las faltas, además es posible que pueda fallar algún diagnosticador. Los diagnosticadores tienen modelos de referencia reducidos del sistema para llevar a cabo el diagnóstico distribuido de faltas en línea en *SED*.

- Se establece una metodología para obtener una distribución de un modelo centralizado diagnosticable. La distribución es óptima porque los módulos tienen el mínimo conjunto de lugares duplicados, los cuales representan la comunicación entre los módulos para detectar y localizar las faltas, esto permite diseñar diagnosticadores distribuidos con comunicación mínima.

Capítulo 2 Diagnóstico Centralizado de Sistemas de Eventos Discretos

Resumen: *En este capítulo se presenta el formalismo de redes de Petri Interpretadas (RPI), para obtener el modelo de SED, el cual permite describir el sistema con estados y eventos parcialmente observables, además de que el modelo incluye las posibles fallas que puedan presentarse en los componentes de un sistema. La propiedad de diagnosticabilidad centralizada es caracterizada para determinar si un modelo RPI es diagnosticable o no, con base a la información estructural del modelo. Finalmente, se presenta el esquema de diagnosticador en línea para detectar y localizar las faltas del sistema.*

2.1. Conceptos Relacionados con redes de Petri

Las redes de Petri (*RP*) [Nissanke, 1997] [Silva, 1985] proveen un formalismo matemático para modelado y análisis de Sistemas de Eventos Discretos, ya que cuentan con una naturaleza gráfica que es conveniente, efectiva y altamente intuitiva que ayuda a la interpretación de sistemas complejos. Además, las *RP* capturan concurrencia, sincronización y no determinismo de los sistemas, permitiendo analizar algunas propiedades tales como vivacidad, ausencia de bloqueos, entre otras.

Las *RP* cubren un rango de aplicaciones diferentes, incluyendo protocolos de comunicación, redes de computadoras, sistemas de manufactura, tráfico urbano, entre otros, además se han usado ampliamente en el estudio de propiedades del comportamiento de sistemas, tales como simulación, evaluación de desempeño y tolerancia a faltas.

Definición 2.1. Una estructura de una red de Petri se representa por la 4-tupla $G=(P,T,I,O)$, donde:

$P = \{p_1,p_2,\ldots,p_n\}$ es un conjunto finito de vértices llamados lugares,
$T = \{t_1,t_2,\ldots,t_m\}$ es un conjunto finito de vértices llamados transiciones,
$I: P \times T \rightarrow \mathbb{Z}^+$ es una función que representa el peso de los arcos que van de lugares a transiciones.
$O: P \times T \rightarrow \mathbb{Z}^+$ es una función que representa el peso de los arcos que van de transiciones a lugares. $\mathbb{Z}^+$ es el conjunto de enteros no negativos.

La función de marcado $M: P \rightarrow \mathbb{Z}^+$ es un mapeo de cada lugar a los enteros no negativos representando el total de marcas (dibujadas como puntos) que contiene cada lugar. El marcado de una *RP* usualmente se expresa como un vector columna, donde el *i-ésimo* componente es el marcado del lugar P_i en un determinado momento.

Definición 2.2. Un sistema de *RP* es un par $N=(G,M_0)$, donde G es la estructura de *RP* y M_0 es la distribución inicial del marcado.

La matriz de incidencia de G es $C = [c_{ij}]$, donde $c_{ij} = O(p_i,t_j) - I(p_i,t_j)$.
Usualmente:

- $^{\bullet}(t_j)$ denota el conjunto de los lugares p_i tal que $I(p_i,t_j) \neq 0$;
- $(t_j)^{\bullet}$ denota el conjunto de los lugares p_i tal que $O(p_i,t_j) \neq 0$;
- $^{\bullet}(p_j)$ denota el conjunto de las transiciones t_i tal que $O(p_i,t_j) \neq 0$; y
- $(p_j)^{\bullet}$ el conjunto de las transiciones t_i tal que $I(p_i,t_j) \neq 0$.

Una transición t_j se habilita en un marcado M_k si y sólo si $\forall p_i \in P \;\; M_k(p_i) \geq I(p_i,t_j)$; una transición habilitada t_j puede ser disparada alcanzando un nuevo marcado M_{k+1} el cual puede ser calculado mediante la ecuación de estado 2.1 de la *RP*:

$$M_{k+1} = M_k + Cv_k \tag{2.1}$$

donde v_k es un vector de m-entradas, cuyas componentes se definen: $v_k(i) = 0$, $i \neq j$, $v_k(j) = 1$ y m es el número de transiciones en la red. Lo anterior también se denota como: $M_k \xrightarrow{t_j} M_{k+1}$.

Ejemplo 2.1. Considere la *RP* de la figura 2.1, la cual se denota por (G, M_0), donde $G = (P, T, I, O)$ y M_0 es la distribución inicial del marcado. El conjunto de lugares de la *RP* es $P = \{p_1, p_2, p_3, p_4, p_5\}$; el conjunto de transiciones es $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$.
La matriz de incidencia que corresponde a esa *RP* es:

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

El marcado inicial de la *RP* es:

$$M_0 = [1 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$$

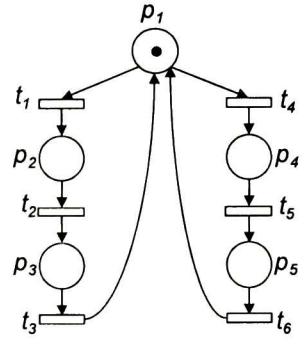

Fig. 2.1. Red de Petri

El ejemplo muestra que en el marcado M_0 se tienen habilitadas las transiciones t_1 y t_4. Si la transición t_1 se dispara, el nuevo marcado alcanzado es:

$$M_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \left(\begin{bmatrix} -1 & 0 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Donde en este nuevo marcado se habilita la transición t_2 y se puede continuar con la ejecución de la red de Petri.

Definición 2.3. El conjunto de alcanzabilidad de una *RP* es el conjunto de todos los posibles marcados alcanzables desde M_0 disparando sólo transiciones habilitadas; este conjunto se denota por $R(G,M_0)$.

Definición 2.4. Una secuencia disparable de (G,M_0) es una secuencia de transiciones $\sigma = t_i t_j ..t_k$ tal que $M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ...M_x \xrightarrow{t_k} ...$

Definición 2.5. Sea $\sigma = t_i t_j ..t_k$ una secuencia finita disparable. El vector de Parikh $\sigma : T \rightarrow (Z^+)^m$ donde σ mapea a cada transición $t \in T$ el número de ocurrencias de t en σ, donde $m = | T |$.

Definición 2.6. El conjunto de todas las secuencias disparables se llama lenguaje de disparo $\mathcal{L}(G,M_0) = \{ \sigma = t_i t_j ..t_k \ldots | M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ...M_x \xrightarrow{t_k} ... \}$.

Definición 2.7. Sea C la matriz de incidencia de *RP* (G,M_0). Un *T-semiflujo* X_i de (G,M_0) es una solución de valores racionales semi-positivos de la ecuación $CX_i = 0$. El soporte del *T-semiflujo* X_i es el conjunto $||X_i|| = \{t_j \mid X_i(t_j) \neq 0\}$.

Definición 2.8. Sea C la matriz de incidencia de *RP* (G,M_0). Un *P-semiflujo* Y_i de (G,M_0) es una solución de valores racionales semi-positivos de la ecuación $Y^T C = 0$. El soporte del *P-semiflujo* Y_i es el conjunto $||Y_i|| = \{p_j \mid Y_i(p_j) \neq 0\}$.

Un *P-semiflujo* representa un componente conservativo en la *RP* donde las marcas permanecen constantes, mientras que un *T-invariante* representa un componente repetitivo donde la secuencia de transiciones que forman parte de un *T-invariante* al momento que se habilitan y se disparan las transiciones se regresa al marcado inicial del *T-invariante*.

Definición 2.9. Sea X_i un *T-semiflujo* de la *RP* (G,M_0) y $||X_i||$ el soporte de X_i, entonces la subred inducida por X_i es $TC_i = (P_i = \cup p_r \in \bullet t_k, p_l \in t_k \bullet; t_k \in ||X_i||, T_i = ||X_i||, I_i, O_i)$, el cual es nombrado *T-componente* inducido por X_i.

Definición 2.10. Sea Y_i un *P-semiflujo* de la *RP* (G,M_0) y $||Y_i||$ el soporte de Y_i, entonces la subred inducida por Y_i es $PC_i = (P_i = ||Y_i||, T_i = \cup t_k \in \bullet p_j, t_l \in p_j \bullet; p_j \in ||Y_i||, I_i, O_i)$, el cual es nombrado *P-componente* inducido por Y_i.

Definición 2.11. Un sifón (cerrojo) es un subconjunto de lugares $S = \{p_1, \ldots, p_s\} \subseteq P$ de una *RP* tal que el conjunto de transiciones de entrada $\bullet S$ se encuentra contenido en el conjunto de transiciones de salida $S\bullet$, es decir $\bullet S \subset S\bullet$.

Ejemplo 2.2. Considere la *RP* de la figura 2.1. Los vectores:

$$X_1 = [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T \text{ y}$$

$$X_2 = [0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1]^T$$

son las soluciones racionales semi-positivas para $CX_i = 0$ ó *T-semiflujos* de la red. Si la secuencia $\sigma = t_1t_2\ t_3$ correspondiente al vector $X_1 = [1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0]^T$ se dispara en la *RP*, el nuevo marcado M_j alcanzado será igual al marcado inicial M_0. De igual manera se cumple que cuando se dispara la secuencia $\sigma = t_4t_5t_6$, correspondiente al vector $X_2 = [0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1]^T$ se alcanza un marcado $M_k = M_0$.

El vector siguiente es la solución racional semi-positiva que cumple con $Y^T C = 0$,

$$Y_1 = [1\ \ 1\ \ 1\ \ 1\ \ 1]^T$$

Nótese que para cualquier marcado alcanzable M_k se tiene que $M_k(p_1) + M_k(p_2) + M_k(p_3) + M_k(p_4) + M_k(p_5) = 1$, esa ecuación indica que para cualquier marcado alcanzable se tiene que en p_1, p_2, p_3, p_4 y p_5 sólo hay una marca. $Y_1 = [1\ \ 1\ \ 1\ \ 1\ \ 1]^T$ es la única componente conservativa ó *P-semiflujo* de la *RP*

Definición 2.12. Una *RP(G,M0)* es cíclica si $\forall M_k \in R(G, M_0)$ se cumple que $\exists \sigma$ tal que $M_k \xrightarrow{\sigma} M_0$.

Definición 2.13. Una *RP(G,M0)* es viva si $\forall M_k \in R(G, M_0)$ y $\forall t \in T$ se cumple que $\exists \sigma$ tal que $M_i \xrightarrow{\sigma} M_j \xrightarrow{t}$.

Definición 2.14. Una *RP(G,M0)* es acotada si existe un número entero positivo D, tal que $\forall M_k \in R(G, M_0)$ se cumple que $\forall p_i, M_k(p_i) \leq D$ entonces *(G,M0)* es D-acotada, si $D = 1$ entonces *(G,M0)* es binaria o segura.

2.1.1. Redes de Petri Interpretadas (*RPI*)

El modelo de *RPI* que se presenta aquí es el propuesto en [Ramírez-Treviño, et al., 2003] que permite representar lenguajes de entrada-salida de *SED*.

Definición 2.15. Una Red de Petri Interpretada (*RPI*) es una 4-tupla $Q = (N, \Sigma, \lambda, \varphi)$ donde:

- $N = (G, M_0)$, donde G es la estructura de *RP* y M_0 es el marcado inicial.
- $\Sigma = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ es el alfabeto de entrada de la red, donde α_i es el i-ésimo símbolo de entrada del alfabeto.
- $\lambda : T \rightarrow \Sigma \cup \{\varepsilon\}$ es una función de etiquetado de las transiciones con las siguientes restricciones: $\forall t_j, t_k \in T, j \neq k$, si $\forall p_i\ I(p_i,t_j) = I(p_i,t_k) \neq 0$ y ambos $\lambda(t_j) \neq \varepsilon$, entonces $\lambda(t_j) \neq \lambda(t_k)$. En este caso ε representa un evento interno del sistema.
- Existe una matriz φ de q x n elementos, tal que, $y_k = \varphi M_k$ es un mapeo del marcado M_k en un vector de observación q-dimensional. La columna $\varphi(\bullet, i)$ es un vector elemental e_h, tal que, el lugar p_i tiene asociado el sensor h; o el vector nulo si el lugar p_i no tiene asociado un sensor. En este caso, el vector elemental e_h es el vector q-dimensional con todas las entradas igual a cero, excepto la entrada h, que es igual a 1. Un vector nulo tiene todas sus entradas igual a cero.

Nota:

- En la tesis *(Q,M0)* se usa en lugar de $Q = (N, \Sigma, \lambda, \varphi)$ para enfatizar la idea de que existe un marcado inicial en la *RPI*.

Definición 2.16. Una transición $t_j \in T$ de una *RPI* está habilitada en el marcado M_k si $\forall p_i \in P$, $M_k(p_i) \geq I(p_i, t_j)$. Si $\lambda(t_j) = a_i \neq \varepsilon$ está presente y t_j está habilitada, entonces t_j debe ser disparada. Cuando una transición habilitada t_j es disparada en un marcado M_k, entonces un nuevo marcado M_{k+1} es alcanzado. Esta idea se representa $M_k \xrightarrow{t_j} M_{k+1}$ y M_{k+1} puede ser calculada usando la parte dinámica de la ecuación de estado: $M_{k+1} = M_k + Cv_j$.

La ecuación de estado de una *RPI* se define como sigue:

$$\begin{aligned} M_{k+1} &= M_k + Cv_k \\ y_k &= \varphi(M_k) \end{aligned} \qquad (2.2)$$

donde C es la matriz de incidencia, v_k es un vector de disparo y $y_k \in (\mathbb{Z}^+)^q$ es la k-ésima columna del vector de salida φ de la *RPI*.

Las funciones λ y φ clasifican a las transiciones y lugares de una *RPI (Q,M0)* de la siguiente manera:

Definición 2.17. Si $\lambda(t_i) \neq \varepsilon$ la transición t_i se dice manipulable. De otra forma es no manipulable. Un lugar $p_i \in P$ es medible si la i-ésima columna del vector columna de φ no es nulo, esto es, $\varphi(\bullet, i) \neq 0$. De otra manera es no medible.

Los conceptos introducidos para *RP* se pueden extender para *RPI*. Las siguientes definiciones relacionan los símbolos de entrada y de salida de *RPI* con las secuencias disparables y secuencias de marcado.

Definición 2.18. Una secuencia de símbolos de entrada-salida de (Q,M_0) es una secuencia $\omega = (\alpha_0, y_0)(\alpha_1, y_1)..(\alpha_n, y_n)$ donde $\alpha_j \in \Sigma \cup \{\varepsilon\}$ y α_{j+1} es la entrada actual de la *RPI*, cuando la salida cambia de y_i a y_{i+1}. Se supone que $\alpha_0 = \varepsilon$, $y_0 = \varphi(M_0)$ y (α_{i+1}, y_{i+1}) son parte de la secuencia cuando:

- (α_i, y_i) pertenece a la secuencia,
- $y_{i+1} \neq y_i$, y
- no existe $y_j \neq y_i$, $y_j \neq y_{i+1}$ ocurriendo después de la ocurrencia de y_j y antes de la ocurrencia de y_{i+1}

Definición 2.19. Sea (Q,M_0) una *RPI*. El conjunto $\Lambda(Q,M_0) = \{\omega \mid \omega$ es una secuencia de símbolos de entrada-salida$\}$ denota el conjunto de todas las secuencias de símbolos de entrada-salida de (Q,M_0). El conjunto de todas las secuencias de entrada-salida de un tamaño mayor que k serán denotados por $\Lambda^k(Q,M_0)$, es decir $\Lambda^k(Q,M_0) = \{\omega \in \Lambda(Q,M_0) \mid |\omega| \geq k\}$.

Definición 2.20. Si $\omega = (\alpha_0, y_0)(\alpha_1, y_1)..(\alpha_n, y_n)$ es una secuencia de símbolos de entrada-salida, entonces el disparo de la secuencia de transición $\sigma \in \mathscr{L}(Q,M_0)$, cuyo disparo realmente genera ω es

denotado por σ_ω. El conjunto de todas las secuencias disparables que generan la palabra ω es definido como $\Omega(\omega) = \{\sigma | \sigma \in \mathcal{L}(Q,M_0) \wedge$ el disparo de σ produce $\omega\}$.

Definición 2.21. El conjunto de las secuencias de símbolos de entrada-salida que llevan a marcados terminales en la *RPI* es denotado por $\Lambda_B(Q,M_0)$, es decir, $\Lambda_B(Q,M_0) = \{\omega \in \Lambda(Q,M_0) \mid \exists\sigma \in \Omega(\omega)$ tal que $M_0 \xrightarrow{\sigma} M_j \wedge si\ M_j \xrightarrow{t_i} entonces\ C(\bullet,t_i) = \vec{0}\}$.

Definición 2.22. Sea $\omega = (\alpha_0, y_0)(\alpha_1, y_1)...(\alpha_n, y_n) \in \Lambda(Q,M_0)$ una secuencia de símbolos de entrada-salida. El conjunto de secuencias de marcados correspondientes a ω es definido como $S_\omega = \{M_0 M_1 ... M_k \mid M_i \in R(Q,M_0) \wedge M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ... \xrightarrow{t_m} M_k \wedge \sigma_\omega = t_i t_j ... t_m \in \Omega(\omega)\}$.

Ejemplo 2.3. Si se definen la función de etiquetado de las transiciones y la función de salida para la *RP* de la figura 2.2 de la siguiente manera:

- $\lambda(t_1) = a$, $\lambda(t_2) = b$, $\lambda(t_3) = c$, $\lambda(t4) = \varepsilon$, $\lambda(t_5) = b$, $\lambda(t_6) = d$.
- $\varphi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$

Entonces, la transición t_4 es la única transición no manipulable y los lugares p_1, p_3 y p_5 son lugares medibles. Los lugares rellenos en color gris representan los lugares no medibles.

El lenguaje de disparo para la red anterior es el siguiente:

$$\mathcal{L}(Q,M_0) = \{t_1, t_1t_2, t_1t_2t_3, t_1t_2t_3t_4, ...\}, \text{ donde,}$$

$$\Lambda(Q,M_0) = \left\{\left(\varepsilon,\begin{bmatrix}1\\0\end{bmatrix}\right),\left(a,\begin{bmatrix}0\\0\end{bmatrix}\right),\left(b,\begin{bmatrix}0\\1\end{bmatrix}\right),\left(c,\begin{bmatrix}1\\0\end{bmatrix}\right),...\right\}$$

$$\Lambda_B(Q, M_0) = \{\}.$$

Si se tiene la siguiente secuencia de símbolos de entrada-salida:

$$\omega = \left(\varepsilon,\begin{bmatrix}1\\0\end{bmatrix}\right),\left(a,\begin{bmatrix}0\\0\end{bmatrix}\right),\left(b,\begin{bmatrix}0\\1\end{bmatrix}\right),\left(c,\begin{bmatrix}1\\0\end{bmatrix}\right)$$

entonces:

$$\Omega(\omega) = \{t_1t_2t_3\} \text{ y}$$

$$S_\omega = \left\{\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}\right\}$$

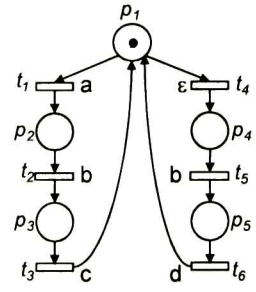

Fig. 2.2. Red de Petri Interpretada

2.1.2. Propiedad de Evento Detectabilidad

En este trabajo la propiedad de evento-detectabilidad es muy importante para el estudio de diagnóstico de faltas a partir de las secuencias disparables en una *RPI*. Si las secuencias pueden ser detectadas usando sólo la salida y la información estructural de la *RPI*, la *RPI* será llamada evento-detectable.

Definición 2.23. Sea (Q,M_0) una *RPI*, una transición t_i de (Q,M_0) es evento-detectable si su disparo puede ser distinguido de otro usando la información de $\omega \in \Lambda(Q,M_0)$.

Definición 2.24. Una $RPI(Q,M_0)$ es evento-detectable si el disparo de cualquier transición puede ser determinado de manera única a partir del conocimiento de la entrada y salida que produce (Q,M_0).

El siguiente lema presentado en [Rivera-Rangel, 2004] define una caracterización de *RPI* evento-detectable.

Lema 2.25. Sea (Q,M_0) una *RPI* viva es evento-detectable si se cumple lo siguiente:

1. $\forall t_i, t_j \in T$ tal que $\lambda(t_i) = \lambda(t_j)$ ó $\lambda(t_i) = \varepsilon$ se cumple que $\varphi C(\bullet,t_i) \neq \varphi C(\bullet,t_j)$ y
2. $\forall t_k \in T$ se cumple que $\varphi C(\bullet,t_k) \neq 0$.

Ejemplo 2.4. Considérese el modelo de *RPI* de la figura 2.2, y considere que:

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

Por lo tanto, el modelo de *RPI* no es evento-detectable ya que existen dos vectores columna correspondientes a las transiciones t_2 y t_5 donde el disparo de esas transiciones no se distingue a partir de la secuencia de símbolos de entrada-salida ya que ambas tienen el mismo símbolo de entrada asociado ($\lambda(t_2) = \lambda(t_5) = b$,) y la misma salida, mientras que las transiciones t_1 y t_4 tienen asociadas la misma salida, sin embargo, los símbolos de entrada son diferentes ($\lambda(t_1) = a$, $\lambda(t_5) = \varepsilon$).

2.2. Modelado del Sistema

El enfoque utilizando en esta tesis se basa en el modelo, entonces en esta sección se describe un Sistema de Evento Discreto utilizando un modelo de *RPI*. Se utiliza la metodología de modelado propuesta en [Ramírez-Treviño, et al., 2007] la cual utiliza como herramienta formal las *RPI*. Esta metodología de modelado construye módulos de *RPI* vivos y binarios para representar el comportamiento de cada uno de los componentes identificados del *SED* y las relaciones entre ellos.

En este trabajo se consideran que los módulos capturan el comportamiento normal y de falta de cada uno componentes del sistema, donde el comportamiento normal está representado por (Q^N,M_0^N) siendo una *RPI* viva y binaria. Analizar la propiedad de vivacidad no puede ser evaluada eficientemente en *RPI*, sin embargo, si se utiliza la metodología de modelado propuesta en [Ramírez-Treviño, et al., 2007] y se preservan las condiciones de [Koh y DiCesare, 1991] en los

circuitos de *RPI* (lo cual se evalúa en tiempo polinomial cuando la metodología previa es usada), entonces la *RPI* (Q^N, M_0^N) generada es viva y segura; también se pueden aplicar otras metodologías para obtener modelos en *RPI* vivas y seguras [Koh y DiCesare, 1991], [Zhou y DiCesare, 1993]. A continuación se describe de manera general los pasos que involucra la metodología propuesta en [Ramírez-Treviño, et al., 2007].

2.2.1. Metodología de modelado

1. Componentes del sistema.- Se identifican y definen los componentes del sistema. Después, se crea un conjunto finito Componentes_Sistema = $\{sc_1, sc_2, \ldots, sc_n\}$ de los componentes identificados. Un componente del sistema puede ser una válvula, un motor, un recurso del sistema, etc.
2. Variables de estado.- Para cada componente identificado del sistema, se necesita establecer las diferentes variables que representen el comportamiento del componente. En otras palabras, se crea un conjunto finito de variables Variables_Estado$_i$= $\{sv^i_1, sv^i_2, \ldots, sv^i_m\}$ asociado a cada componente $sc_i \in$ Componentes_Sistema. Las variables pueden representar la posición, velocidad, voltaje, etc. de cada componente del sistema, o puede representar un descriptor de tareas (para una instancia de una máquina de estado). Existe al menos una variable de estado para cada componente del sistema.
3. Conjunto de valores.- Para cada variable de estado $sv^i_j \in$ Variables_Estado$_i$, se establece el conjunto de valores posibles $Value_{sv^i_j} = \{val^y_1, val^y_2, \ldots, val^y_p\}$ de cada variable de estado sv^i_j. También se consideran valores de falta asociados a cada componente. Por ejemplo, la variable "posición_válvula" puede tomar cuatro valores: "Abierto", "Cerrado", "ErrorAlAbrirse" y "ErrorAlCerrarse"
4. Codificación.- Los valores en cada conjunto $Value_{sv^i_j}$ deben ser representados en términos de marcados de redes de Petri. Esto puede ser realizado fácilmente si se utilizan lugares binarios. Para cada $sv^i_j \in$ Variables_Estado$_i$ se crea un conjunto $P_{sv^i_j} = \{p^y_1, p^y_2, \ldots, p^y_n\}$ de lugares tal que $|Value_{sv^i_j}| = |P_{sv^i_j}|$ El marcado de estos lugares es binario y mutuamente excluyente. Entonces, si $M(p^{ij}_z) = 1$ significa que la variable sv^i_j toma el valor val^y_z. Debido a la existencia de valores de falta, el conjunto de lugares se particiona es los subconjuntos $P^F_{sv^i_j}$ y $P^{\lambda}_{sv^i_j}$, representando el comportamiento de falta y normal respectivamente.
5. Modelado de eventos.- Para cada par de valores val^y_m, val^y_n tal que la variable de estado sv^i_j pudiera cambiar del valor val^y_m al valor val^y_n, una transición t^y_{mn} debe ser creada. Entonces, se agregan dos arcos: uno que va desde el lugar p^y_m a la transición t^y_{mn} y otro de la transición t^y_{mn} al lugar p^y_n.
6. Marcado inicial.- El marcado inicial se define como: $M_0(p^y_m)$=1 si el valor inicial de la variable sv^i_j es val^y_m y $M_0(p^y_m)$=0 de otra manera.
7. Salida.- La salida de este algoritmo es un conjunto de módulos de *RP* aislados, cada módulo modelando el comportamiento de la variable de estado sv^i_j.
8. Ejecución de la composición síncrona y permisiva a través de los módulos *RPI* para obtener el modelo global del sistema, ver procedimiento presentado en [Ramírez-Treviño, et al. 2007].

Ejemplo 2.6. Considere la celda de producción de la figura 2.3. Esta celda está compuesta por 3 máquinas (nombradas M1, M2 y M3), tres robots (R1, R2 y R3), dos almacenes de entrada (I1 y I2) y dos almacenes de salida (O1 y O2). En la celda se pueden producir dos tipos de productos. El plan

de trabajo para el producto 1 es el siguiente: el robot R_1 toma una parte sin procesar del almacén de entrada I_1 y lo transporta a la máquina M_1, posteriormente el robot R_2 toma el producto de la máquina M_1 y lo transporta a la máquina M_3 donde finalmente se termina de realizar, una vez que se termina de fabricar el producto 1 el robot R_3 lo transporta al almacén de salida O_1. El plan de trabajo para el producto 2 es el siguiente: el robot R_1 toma una parte sin procesar del almacén de entrada I_2 y lo transporta a la máquina M_2, posteriormente el robot R_2 toma el producto de la máquina M_2 y lo transporta a la máquina M_3 donde finalmente se termina de realizar, una vez que se termina de fabricar el producto 2 el robot R_3 lo transporta al almacén de salida O_2.

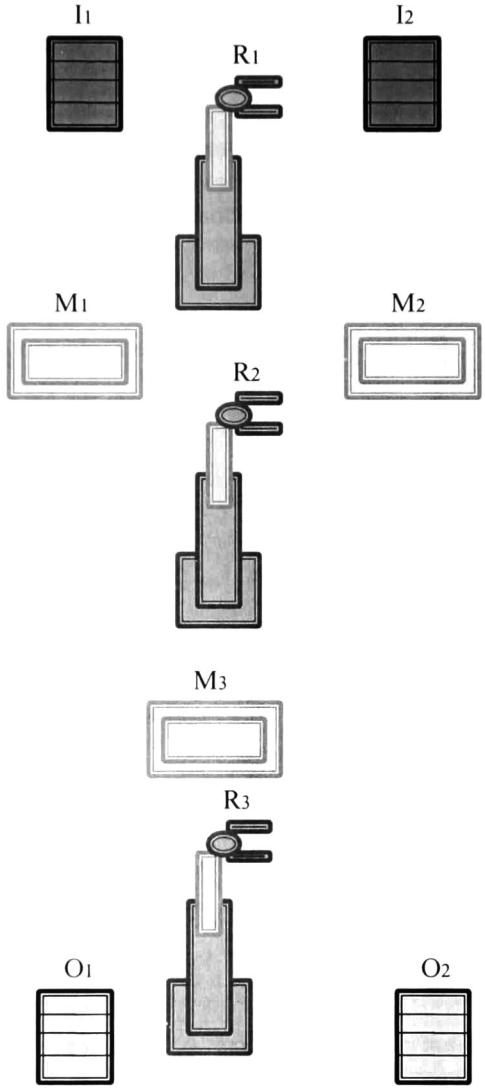

Fig. 2.3. Celda de Producción

En base a la metodología propuesta anteriormente, se identifican diez componentes: I_1, I_2, R_1, R_2, R_3, M_1, M_2, M_3, O_1, O_2. Las variables de estado involucradas son: "tareas_I_1", "tareas_I_2", "tareas_R_1", "tareas_R_2", "tareas_R_3", "tareas_M_1", "tareas_M_2", "tareas_M_3", "tareas_O_1" y "tareas_O_2" para describir respectivamente el comportamiento de I_1, I_2, R_1, R_2, R_3, M_1, M_2, M_3, O_1, O_2.

El rango de cada una de las variables de estado es:

- Rango_tareas_I_1 = {ocupado, vacío},
- Rango_tareas_I_2 = {ocupado, vacío},
- Rango_tareas_R_1 = {disponible, cargando_parte_I_1, transportando_parte_I_1, incorporando_parte_I_1_en_M_1, cargando_parte_I_2, transportando_parte_I_2, incorporando_parte_I_2_en_M_2},
- Rango_tareas_R_2 = {disponible, cargando_parte_procesada_de_M_1, cargando_parte_procesada_de_M_2, incorporando_parte_de_M_1_en_M_3, incorporando_parte_de_M_2_en_M_3},
- Rango_tareas_R_3 = {disponible, transportando_parte_terminada_de_I_1, dejando_parte_I_1_en_O_1, transportando_parte_terminada_de_I_2, dejando_parte_I_2_en_O_2},
- Rango_tareas_M_1 = {desocupado, procesando_parte_de_I_1, esperando_robot_R_2},
- Rango_tareas_M_2 = {desocupado, procesando_parte_de_I_2, esperando_robot_R_2},
- Rango_tareas_M_3 = {desocupado, terminando_parte_procesada_de_I_1, terminando_parte_procesada_de_I_2, esperando_robot_R_3_parte_de_I_1, esperando_robot_R_3_parte_de_I_2},
- Rango_tareas_O_1 = {ocupado, vacío},
- Rango_tareas_O_2 = {ocupado, vacío}.

Los conjuntos anteriores son codificados usando los siguientes conjuntos de lugares:

- P_tareas_I_1 = {p_1, p_2},
- P_tareas_I_2 = {p_3, p_4},
- P_tareas_R_1 = {p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}},
- P_tareas_R_2 = {p_{18}, p_{19}, p_{20}, p_{21}, p_{22}},
- P_tareas_R_3 = {p_{28}, p_{29}, p_{30}, p_{31}, p_{32}},
- P_tareas_M_1 = {p_{12}, p_{13}, p_{14}},
- P_tareas_M_2 = {p_{15}, p_{16}, p_{17}},
- P_tareas_M_3 = {p_{23}, p_{24}, p_{25}, p_{26}, p_{27}},
- P_tareas_O_1 = {p_{33}, p_{34}},
- P_tareas_O_2 = {p_{35}, p_{36}}.

Los módulos de *RP* presentados en la figura 2.4 modelan el comportamiento normal de los componentes identificados en el sistema.

Las funciones de etiquetado para llevar a cabo la composición son las siguientes:

- $tLab(t_i) = x_i$, $tLab(t'_1) = x'_1$, $tLab(t_5) = x'_1$, lo anterior indica las transiciones t_1 y t_5 se etiquetan igual, ya que se relacionarán de alguna manera cuando se apliquen las composiciones entre los modelos de los componentes, lo mismo sucede para el resto de las etiquetas definidas.

- $pLab(p_i) = z_i$.

Una vez que se aplica la composición y producto síncrono de los componentes del sistema, se obtiene el modelo global *RPI* mostrado en la figura 2.5.

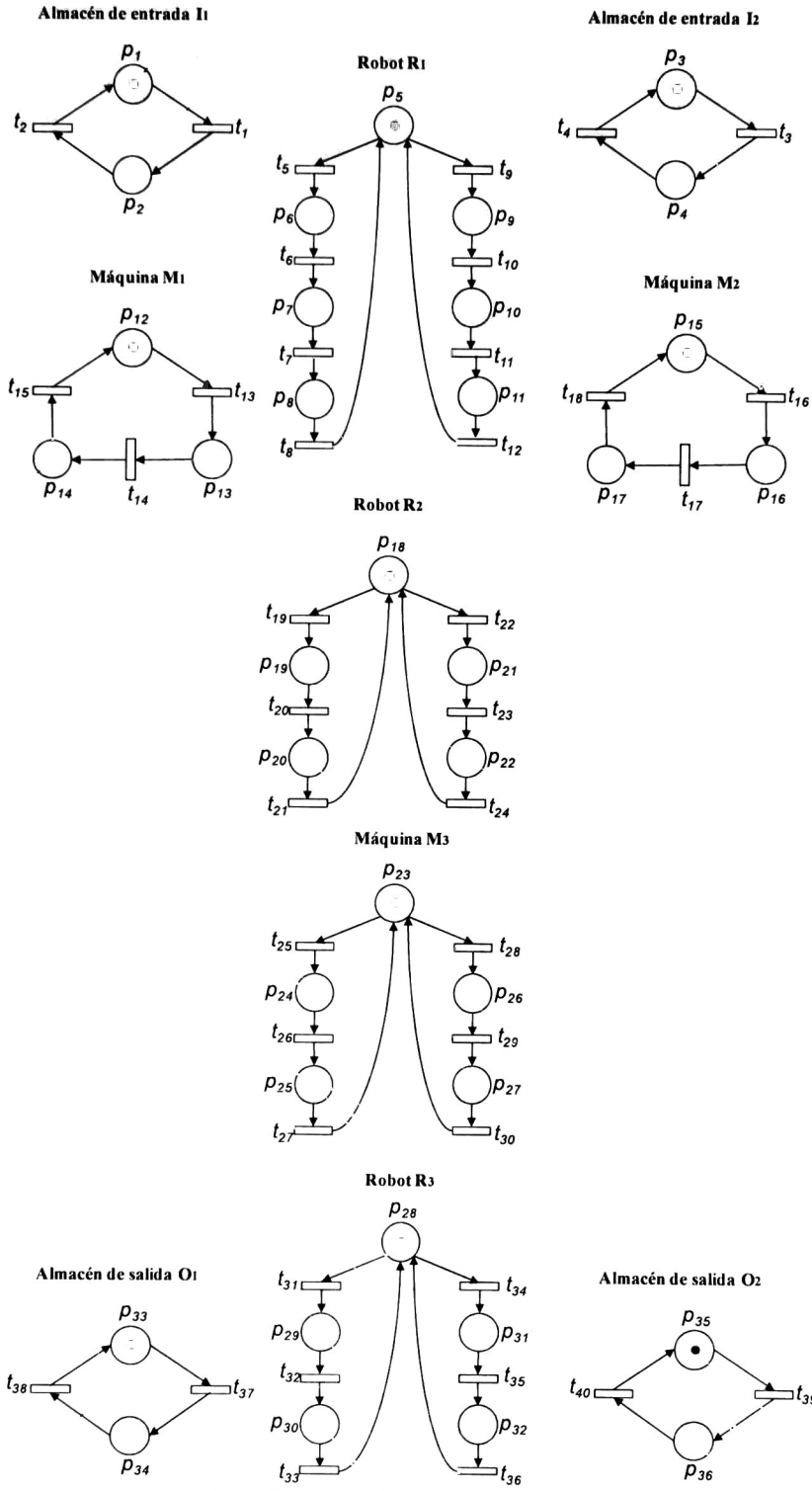

Fig. 2.4. Módulos de RPI obtenidos

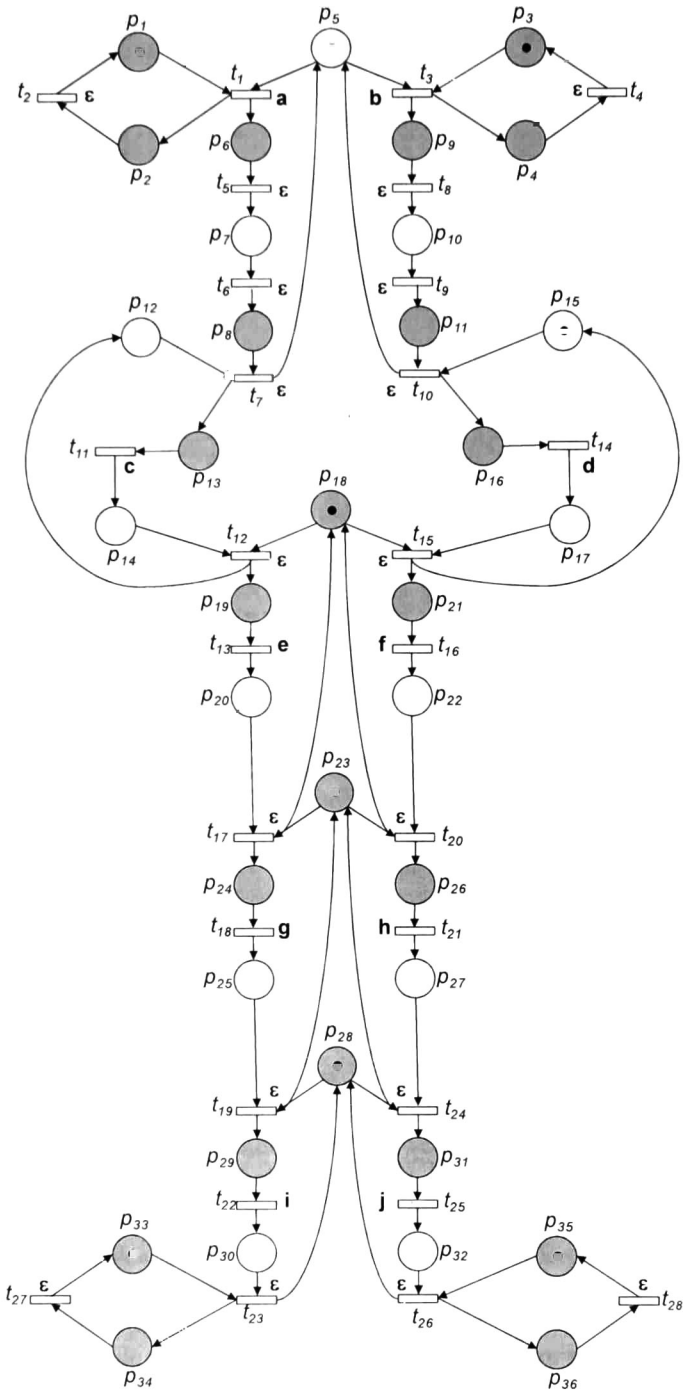

Fig. 2.5. Modelo Global de una Celda de Producción

A continuación se proponen las funciones λ y φ para el modelo de la figura 2.5.

- $\lambda(t_1)$ = a, $\lambda(t_3)$ = b, $\lambda(t_{11})$ = c, $\lambda(t_{14})$ = d, $\lambda(t_{13})$ = e, $\lambda(t_{16})$ = f, $\lambda(t_{18})$ = g, $\lambda(t_{21})$ = h, $\lambda(t_{22})$ = i, $\lambda(t_{25})$ = j, para el resto de transiciones $\lambda(t_i)$ = ε. En la tabla 2.3 se especifica el significado de cada evento manipulable.

$$\varphi = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Debido a que los lugares p_5, p_7, p_{10}, p_{12}, p_{14}, p_{15}, p_{17}, p_{20}, p_{22}, p_{25}, p_{27}, p_{30}, p_{32} son lugares medibles la matriz de φC es la siguiente:

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

En las columnas de la matriz φC se observa que las transiciones: t_2, t_4, t_{27} y t_{28} son transiciones no evento-detectables, aunque las transiciones t_1 y t_3 se confunden a la salida, la entrada es manipulable y tienen asociadas símbolos diferentes por lo tanto es posible distinguir su disparo, el resto de las transiciones son evento-detectables según el lema 2.25. Además del comportamiento normal, el modelo del sistema debe incluir el comportamiento de falta. A continuación, se proponen las metodologías para modelar dos tipos de faltas: operacionales y permanentes, éstas se definen y describen en las siguientes subsecciones.

2.2.2. Modelado de Faltas Operacionales

Definición 2.26. Una falta operacional se presenta en el sistema cuando hay una evolución de un estado a otro, siendo ambos estados válidos, pero no se realiza la ejecución de una o más tareas especificadas en el comportamiento normal.

Las faltas operacionales son agregadas en el comportamiento de la *RPI* de la siguiente forma: cuando existe una secuencia de transiciones $\sigma = t_i t_j \ldots t_k$ representando tareas *i, j, ...,k* y la secuencia se omite, entonces se agrega una transición no manipulable t_f y los arcos correspondientes de: ${}^\bullet t_i$ a t_f y de t_f a $t_k{}^\bullet$. Si la transición t_f se dispara entonces se omite la ejecución de σ. Finalmente, el modelo obtenido representará el comportamiento normal y las faltas operacionales del *SED*. Los estados del grafo de alcanzabilidad con las faltas operacionales se incluyen en $R(Q^N, M_0^N)$

Ejemplo 2.7. Considérese el sistema de manufactura propuesto en el ejemplo 2.6 y supóngase que pueden presentarse las siguientes faltas operacionales: cuando el robot R_1 se encuentra en el estado de cargando_parte_I_1 ó cargando_parte_I_2. Por lo tanto, la falta operacional dirige al sistema a un estado no deseado a partir del estado normal. El modelo de *RPI* que representa el comportamiento

normal y de faltas operacionales del sistema de manufactura se muestra en la figura 2.6. Las funciones de etiquetado de transiciones asociadas a las faltas de control son las siguientes: $\lambda(t_{ofl}) = \varepsilon$, $\lambda(t_{of2}) = \varepsilon$, se puede observar que estas transiciones no son evento-detectable.

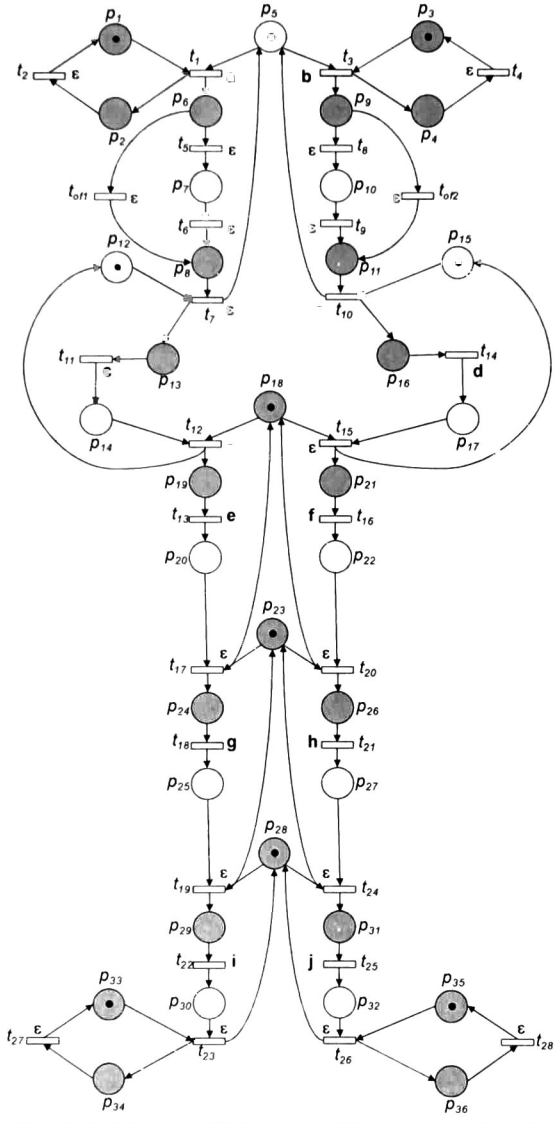

Fig. 2.6. Modelo Global con faltas operacionales

2.2.3. Modelado de Faltas Permanentes

Definición 2.27. Una falta permanente se presenta en el sistema cuando un componente detiene su ejecución en el sistema y no puede continuar su funcionamiento.

Las faltas permanentes son incluidas en el sistema aplicando el siguiente procedimiento: para cada lugar p_i^N representando una operación en la cual una falta permanente puede ocurrir, agregar una transición no manipulable t_f, y un lugar de falta p_j^f, y los arcos (p_i^N, t_f) y (t_f, p_i^f). El nuevo lugar de falta p_i^f debe ser etiquetado con el mismo símbolo de salida que p_i^N para manifestar

que la falta no puede ser detectada directamente de la información de salida (de otra forma la detección de la falta sería trivial).

Ejemplo 2.8. Considere nuevamente la celda de manufactura y supóngase que las faltas permanentes pueden presentarse en los siguientes casos:

- Cuando M1 está en el estado de procesando_parte_de_I1.
- Cuando M2 está en el estado de procesando_parte_de_I2.
- El robot R2 se encuentra en: cargando_parte_procesada_de_M1 ó bien, cargando_parte_procesada_de_M2.
- La máquina M3 se encuentra en el estado: terminando_parte_procesada_de_I1 ó bien, terminando_parte_procesada_de_I2.
- El robot R3 se encuentra en: transportando_parte_terminada_de_I1 ó bien, transportando_parte_procesada_de_I2.

El modelo de *RPI* que representa las faltas se muestra en la figura 2.7. Las siguientes funciones de etiquetado son propuestas para las transiciones que representan faltas permanentes: $\lambda(t^i_{pf}) = \varepsilon$, i = 1…8. Cada lugar de falla tiene asociada la misma salida que su lugar predecesor.

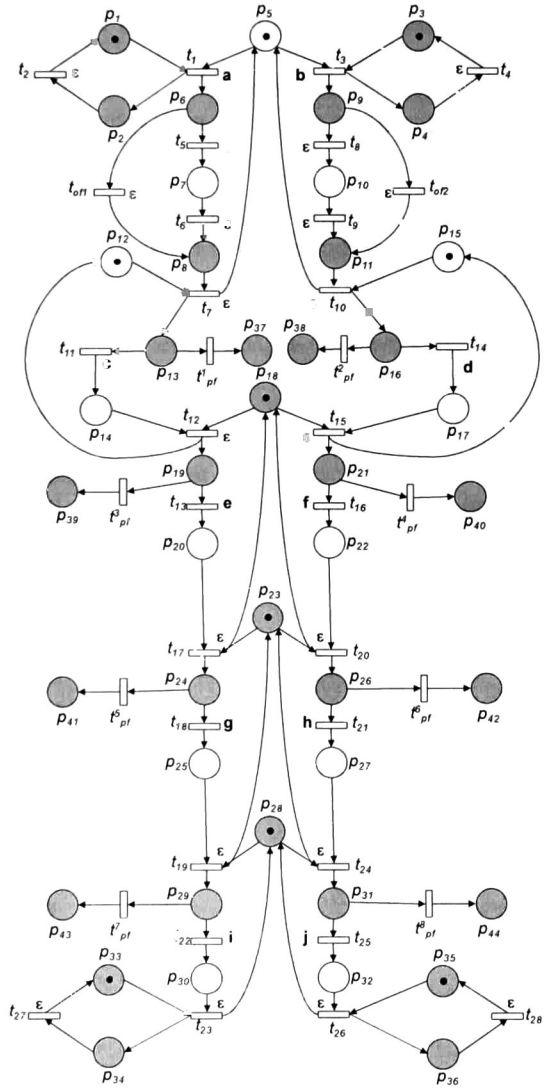

Fig. 2.7. Modelo Global con el comportamiento normal y de falta

2.3. Definición de Diagnosticabilidad

La caracterización de *RPI* diagnosticables entrada-salida se basa en la partición del grafo de alcanzabilidad $R(Q,M_0)$ en marcados de falta y normales, los cuales, deben ser distinguibles del conjunto de marcados alcanzados por la *RPI*. Un marcado M_x donde ocurrió una falta, es distinguible si no existe una secuencia de transiciones $\sigma_l = t_j...t_k$ que se disparen infinitamente alcanzando un marcado M_z, si la secuencia σ_l está presente entonces no es posible distinguir si el sistema se encuentra en el marcado de falta M_x o en el marcado M_z. Si el comportamiento de secuencia infinita no se presenta en el sistema modelado, entonces los marcados de falta pueden ser distinguibles. Si en el marcado M_x se presentó una falta operacional t_{ofi} que dirige al sistema a un marcado M_j entonces debe existir una secuencia de transiciones normales $\sigma_k = t_h...t_m$ en el marcado M_x que al dispararse llegan al mismo marcado M_j, y al menos una transición t_i de la secuencia σ_k es evento-detectable, por lo tanto, si no se disparó la transición t_i al llegar al marcado M_j entonces en el marcado alcanzado ocurrió la falta operacional t_{ofi}, en este caso el sistema continúa su evolución sin haber realizado la secuencia σ_k. Para distinguir un marcado de falta M_j donde se presentó una falta permanente t^j_{pf}, la secuencia $\sigma_k = t_h...t_m$ dirige al sistema a un marcado M_k, si la transición evento-detectable t_i de la secuencia no se dispara, entonces el sistema no alcanza el marcado M_k, el sistema se encuentra en un marcado de bloqueo de falta M_j donde la transición t_i no se dispara, sin embargo, cuando la falta que se presenta en el sistema es permanente entonces el sistema detiene su evolución en número finito de pasos.

Ejemplo 2.9. Considere la *RPI* de la figura 2.8, donde existen dos faltas permanentes $T^{PF} = \{t^1_{pf}, t^2_{pf}\}$ y una falta operacional $T^{OF} = \{t_{of1}\}$. El correspondiente grafo de alcanzabilidad presentado en la figura 2.9, muestra que después de la presencia de la falta permanente t^2_{pf} del sistema será inevitable un bloqueo, el caso contrario sucede, cuando la transición t^1_{pf} se dispara, el sistema puede ejecutar indefinidamente la secuencia t_6t_7 posteriormente; es equivalente al ciclo f_i-indeterminado definido por [Sampath, et al., 1996]. Además, se observa en el grafo que las transiciones t_3 y t_{of1} se encuentran habilitadas en un marcado M_j, entonces, si el disparo de la transición evento-detectable t_3 no se detecta y se llega a un marcado M_k, el sistema evoluciona y al momento de dispararse t_1 podemos detectar que M_k es un marcado de falta ya que t_3 no se disparó entonces ocurrió el disparo de la transición correspondiente a la falta operacional t_{of1}. En este trabajo se proponen técnicas de análisis para detectar estos comportamientos, las cuales son técnicas construidas a partir de la estructura del modelo.

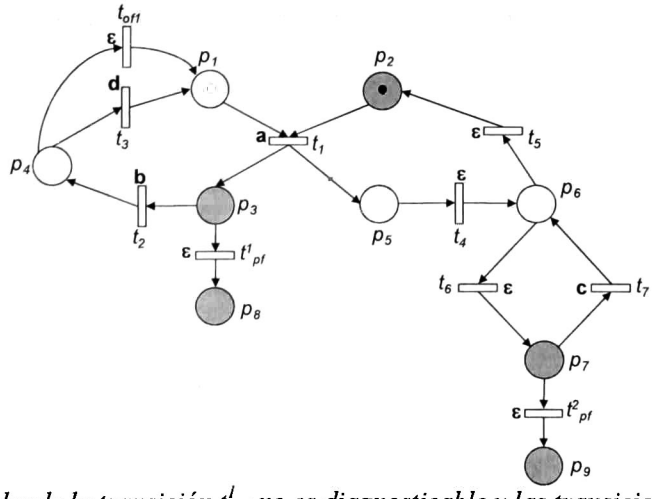

Fig. 2.8. RPI donde la transición t^1_{pf} no es diagnosticable y las transiciones t_{of1} y t^2_{pf} son diagnosticables

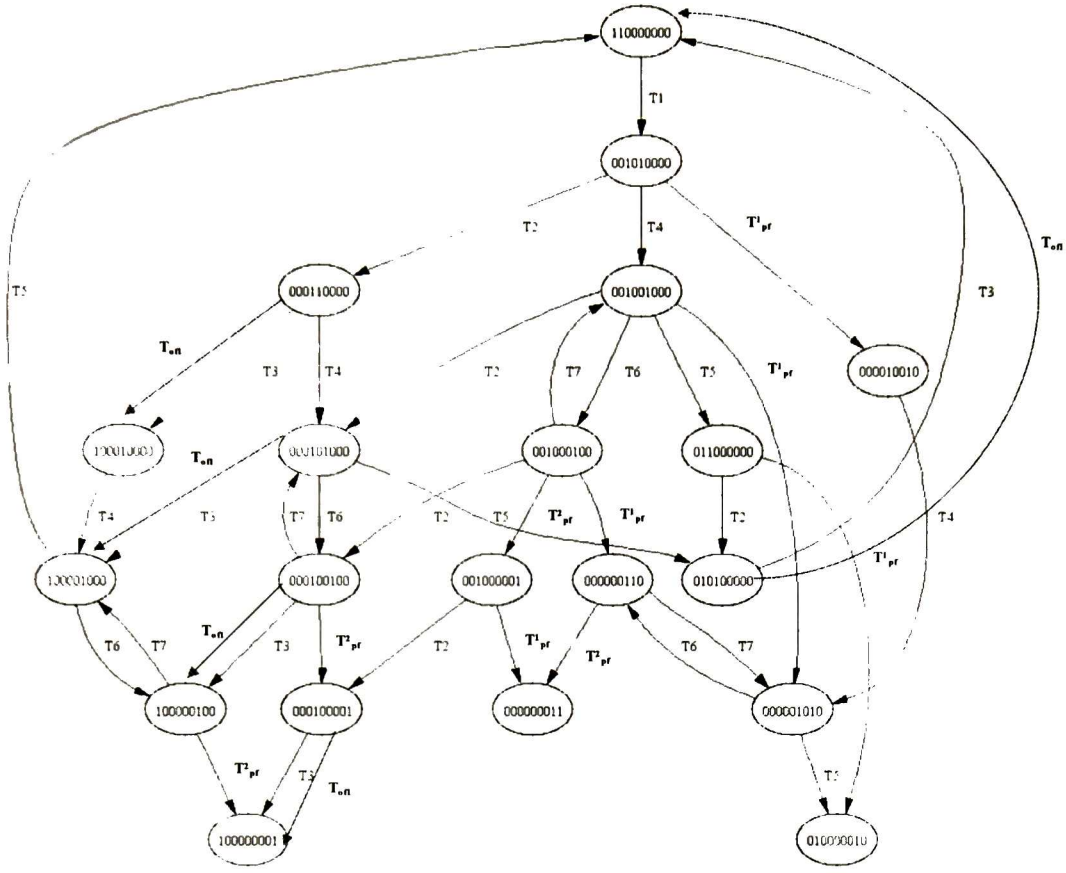

Fig. 2.9. Grafo de alcanzabilidad de la figura 2.8.

En un modelo del sistema utilizando *RPI* el conjunto de nodos se particiona en dos subconjuntos:

- $P = P^N \cup P^{PF}$ donde P^N son los lugares que codifican estados normales de la *RPI* y P^{PF} el conjunto de lugares codificando lugares de falta permanente.

- $T=T^N \cup T^{PF} \cup T^{OF}$ representando el conjunto de transiciones normales, de falta permanente y de falta operacional, donde $T^{PF} = {}^\bullet P^{PF}$

El comportamiento normal embebido (Q^N,M_0^N) de (Q,M_0) es la *RPI* incluida en (Q,M_0) cuando no se consideran los conjuntos: P^{PF}, T^{PF} y T^{OF} En (Q^N,M_0^N), el conjunto de lugares es: $P^N = P - P^{PF}$, el conjunto de transiciones es: $T^N = T - (T^{PF} \cup T^{OF})$ y el conjunto de arcos de (Q^N,M_0^N) es: $A^N = ((P^N \times T^N) \cup (T^N \times P^N)) \cap A$, donde $A = \{(p_i, t_j) \mid p_i \in P, t_j \in T$ y $I(p_i, t_j) = 1\} \cup \{(t_j, p_i) \mid p_i \in P, t_j \in T$ y $O(p_i, t_j) = 1\}$.

Ejemplo 2.10. Considere el modelo global que se muestra en la figura 2.7, se obtienen los siguientes conjuntos: $P^N = \{p_1, \ldots, p_{36}\}$, $P^{PF} = \{p_{37}, \ldots, p_{44}\}$, $T^N = \{t_1, \ldots t_{28}\}$, $T^{PF} = \{t^1_{pf}, \ldots, t^8_{pf}\}$ y $T^{OF} = \{t_{of1}, t_{of2}\}$.

Definición 2.28. Sea (Q,M_0) una *RPI*. (Q,M_0) es diagnosticable entrada-salida en $k < \infty$ pasos si y sólo si usando $\omega \in \Lambda^k(Q,M_f) \cup \Lambda_B(Q,M_f)$ y la estructura de (Q,M_0) son suficiente para distinguir la ocurrencia de faltas en el *SED*.

La definición anterior es equivalente a la que está presentada en [Sampath, et al., 1996] desde el punto de vista de *RPI*. De hecho, si un ciclo F_i – indeterminado aparece en el grafo de alcanzabilidad ó un marcado de bloqueo no determinado (son aquellos marcados con una o más etiquetas de faltas ó con etiquetas normales ó de falta) aparece, entonces la *RPI* no es diagnosticable entrada-salida. Lo inverso también es cierto. La propiedad de diagnosticabilidad en *RPI* presentada en este trabajo asegura que en el sistema podrán ser detectadas las faltas cuando no se presenten los casos siguientes: 1) existe una secuencia infinita de disparo de transiciones que dirige a marcados de bloqueo no determinados donde no es posible distinguir si ocurre ó no una falta permanente en el sistema, 2) existen varias secuencias de disparo de transiciones no evento-dectectables, algunas secuencias pertenecen a un comportamiento normal y existen secuencias donde se disparan transiciones de faltas operacionales, sin embargo, como no existe información que permita distinguir la secuencia donde ocurren faltas operacionales de las otras secuencias, entonces no es posible distinguir si el sistema se encuentra en estado normal o con falta.

Definición 2.29. Sea (Q,M_0) una *RPI*, P^N el conjunto normal de lugares. Sea t_i una falta operacional o permanente, esto es, $t_i \in (T^{PF} \cup T^{OF})$. El conjunto de lugares de riesgo de t_i es: $P_i^R = \{p_k \mid p_k \in {}^{\bullet}t_i\}$. El conjunto de lugares post-riesgo es $P_i^{PR} = \{p_k \mid p_k \in t_i^{\bullet}$, si t_i es una falta operacional, $p_k \in ({}^{\bullet}t_i)^{\bullet\bullet} \cap P^N$ si t_i es una falta permanente$\}$. El conjunto de transiciones pre-riesgo de t_i es $T_i^R = \{t_k \mid t_k \in {}^{\bullet}P_i^R \cap T^N\}$, y el conjunto de transiciones post-riesgo es: $T_i^{PR} = \{t_k \mid t_k \in P_i^{PR\bullet} \cap T^N$ si t_i es una falta operacional, ó $t_k \in P_i^{R\bullet} \cap T^N$ si t_i es una falta permanente$\}$. La secuencia de transiciones de un camino generado por una falta operacional t_i es: $T_i^{Path} = \{\sigma = t_i t_j \ldots t_k \mid \sigma$ es una secuencia de transiciones omitida cuando se dispara $t_i\}$.

Ejemplo 2.11. Considere el modelo global que se muestra en la figura 2.7. La falta operacional t_{of1}, tiene los siguientes conjuntos: lugares de riesgo, post-riesgo, transiciones pre-riesgo, post-riesgo y secuencia de transiciones de un camino generado, esto es, $P_{of1}{}^R = \{p_6\}$, $P_{of1}{}^{PR} = \{p_8\}$, $T_{of1}{}^R = \{t_1\}$, $T_{of1}{}^{PR} = \{t_7\}$ y $T_{of1}{}^{Path} = \{t_5t_6\}$. Para la falta operacional t_{of2}, existen: $P_{of2}{}^R = \{p_9\}$, $P_{of2}{}^{PR} = \{p_{11}\}$, $T_{of2}{}^R = \{t_3\}$, $T_{of2}{}^{PR} = \{t_{10}\}$ y $T_{of2}{}^{Path} = \{t_8t_9\}$. Las 8 faltas permanentes tienen los conjuntos: $P_1{}^R=\{p_{13}\}$, $P_1{}^{PR}=\{p_{14}\}$, $T_1{}^R=\{t_7\}$, $T_1{}^{PR}=\{t_{11}\}$, $P_2{}^R=\{p_{16}\}$, $P_2{}^{PR}=\{p_{17}\}$, $T_2{}^R=\{t_{10}\}$, $T_2{}^{PR}=\{t_{14}\}$, $P_3{}^R=\{p_{19}\}$, $P_3{}^{PR}=\{p_{20}\}$, $T_3{}^R=\{t_{12}\}$, $T_3{}^{PR}=\{t_{13}\}$, $P_4{}^R=\{p_{21}\}$, $P_4{}^{PR}=\{p_{22}\}$, $T_4{}^R=\{t_{15}\}$, $T_4{}^{PR}=\{t_{16}\}$, $P_5{}^R=\{p_{24}\}$, $P_5{}^{PR}=\{p_{25}\}$, $T_5{}^R=\{t_{17}\}$, $T_5{}^{PR}=\{t_{18}\}$, $P_6{}^R=\{p_{26}\}$, $P_6{}^{PR}=\{p_{27}\}$, $T_6{}^R=\{t_{20}\}$, $T_6{}^{PR}=\{t_{21}\}$, $P_7{}^R=\{p_{29}\}$, $P_7{}^{PR}=\{p_{30}\}$, $T_7{}^R=\{t_{19}\}$, $T_7{}^{PR}=\{t_{22}\}$, $P_8{}^R=\{p_{31}\}$, $P_8{}^{PR}=\{p_{32}\}$, $T_8{}^R=\{t_{24}\}$, $T_8{}^{PR}=\{t_{25}\}$.

2.4. Caracterización de Diagnosticabilidad Centralizada en RPI

Para caracterizar la diagnosticabilidad en una *RPI*, se necesita evaluar todos los ciclos F_i-*indeterminados* en el grafo de alcanzabilidad. Desafortunadamente, la obtención de un grafo de alcanzabilidad es un problema NP-completo. Este trabajo se basa en el análisis de propiedades estructurales de las *RPI* (mediante algoritmos eficientes) para determinar la diagnosticabilidad de *SED*.

Para evitar la construcción del grafo de alcanzabilidad, se usan las transiciones post-riesgo, de hecho en el caso de faltas permanentes, estas transiciones pertenecen a cualquier secuencia de disparo de transiciones del sistema, entonces el disparo de las transiciones post-riesgo asegurarán

que una falta no se presentó en el sistema, de lo contrario habrá ocurrido alguna falta. Si $t_i \in T^{OF}$ entonces se considera que antes de que las transiciones post-riesgo se disparen se debe disparar una transición del conjunto T_i^{Path} y si se omite, entonces una falta operacional se presenta. La idea se basa en analizar a los T-componentes inducidos por los T-semiflujos que comparten lugares con todos los P-componentes inducidos por los P-semiflujos donde se encuentran los lugares de riesgo. Cuando se agregan transiciones de falta permanentes se tienen sifones, por lo tanto, es necesario definir un tipo de distancia que muestre la relación que existe entre los sifones y T-semiflujos.

A continuación se presentan los conceptos de distancia relativa [Ramírez-Treviño, et al., 2007] y sifones de una *RPI* para determinar cuándo una transición post-riesgo pertenece a todas las secuencias de la red.

Definición 2.30. Sea (Q,M_0) una *RPI* segura, la distancia relativa $D_R(t_i,t_j)$ entre cualquier par de transiciones $t_i, t_j \in T$, es el número máximo de disparos que puede tener t_j de t_i, cuando una marca permanece en los lugares ${}^{\bullet}t_i$ (esto es, la marca no puede ser usada para disparar la transición t_i).

Definición 2.31. Sea (Q,M_0) una *RPI*, la distancia relativa máxima $D_H(t_i,t_j)$ entre cualquier par de transiciones $t_i, t_j \in T$ es $D_H(t_i,t_j) = \max\{D_R(t_i,t_j), D_R(t_j,t_i)\}$.

El problema de caracterizar la diagnosticabilidad entrada-salida de las faltas requiere la obtención de distancia relativa máxima. La obtención de la distancia es un problema complejo, sin embargo, existen condiciones estructurales para algunas subclases que pueden ser explotadas para determinar polinomialmente la distancia relativa máxima en una amplia clase de *RPI*. La siguiente proposición presentada en [Ramírez-Treviño, et al., 2007] caracteriza un caso cuando la distancia relativa máxima entre cualquier par de transiciones siempre es finita en una *RPI*.

Proposición 2.32. Sea (Q,M_0) una *RPI* segura, donde el comportamiento normal (Q^N,M_0^N) es vivo y fuertemente conexo. Sea $X = \{X_1, \ldots, X_\tau\}$ es el conjunto de *T-semiflujos mínimos* de (Q,M_0). Si los *T-semiflujos* $X_r, X_s \in X$, tal que, $\| X_r \|$ y $\| X_s \|$ comparten transiciones t_r^a, t_s^a con el mismo sifón S de (Q,M_0), entonces $\forall t_r^a \in \| X_r \|, \forall t_s^a \in \| X_r \|, D_H(t_r^a, t_s^a) < \infty$.

Demostración. Sea $t_s \in S$, $t_r^a \in \| X_r \|$ y $t_s^a \in \| X_r \|$. Suponga que el sifón S no está marcado, por lo tanto ninguna de sus transiciones puede ser disparada, especialmente las transiciones t_s, t_r^a, t_s^a y $D_H(t_r^a, t_s) < \infty$ y $D_H(t_s^a, t_s) < \infty$. Si t_r^a no puede ser disparada, entonces todos las transiciones del *T-semiflujo* $\| X_r \|$ no serán disparadas. Es equivalente decir que todas las transiciones del sifón S no pueden ser disparadas, especialmente aquella transición compartida con X_r. Es equivalente decir que el sifón S no está marcado, entonces $D_H(t_r^a, t_s^a) < \infty$.

La siguiente definición de área de influencia para una falta permanente u operacional se deriva de la proposición anterior.

Definición 2.33. Sea (Q,M_0) una *RPI* segura y t_i una falta. Sea P_i^{PR} el conjunto de lugares post-riesgo de t_i. El área de influencia N_i de t_i es la red formada por la unión de todos los *P-componentes* Θ_k generados por los *P-semiflujos* tal que $\Theta_k(p) \neq 0$, donde $p \in P_i^{PR}$ y todos los *T-componentes* generados por los *T-semiflujos* compartiendo transiciones con los *P-componentes* Θ_k.

Es importante señalar que si un marca permanece en un lugar del área de influencia, entonces los *P-componentes* pueden ser interpretados como sifones no marcados, debido a que la marca no está disponible en los *P-componentes;* entonces el área de influencia de t_f incluye todas las transiciones t_k donde $D_H(t_f, t_k) < \infty$.

Proposición 2.34. Sea (Q^N, M_0^N) una *RPI* segura, viva, fuertemente conexa, donde las faltas son modeladas como en las secciones 2.2.1 y 2.2.2, donde para cualquier transición de falta t_f, todas las transiciones en T_f^R, T_f^{PR} son evento detectables, y al menos una transición por cada $\sigma \in T_f^{Path}$ es evento-detectable. Sea N_f el área de influencia generada por la falta t_f. Si N_f es (Q^N, M_0^N) y $\forall t_k \in T_f^{PR}$, $|^\bullet(t_k)| = \{p_i^N\}$, donde $|^\bullet(t_k)| = 1$, y $\lambda(t_k) \neq \varepsilon$ entonces (Q, M_0) es diagnosticable entrada-salida.

Demostración. Debido a que (Q^N, M_0^N) es vivo, entonces es vivo por lugares [Desel y Esparza, 1995]. Significa que cualquier lugar $p_i \in P_i^R$ será eventualmente marcado. Sin pérdida de generalidad, se asume que en un marcado M_k, $M_k(p_i) = 1$ y que $t_f \in p_i^\bullet$. Significa que la transición de falta t_f está habilitada en M_k. Si se supone que la falta t_f se dispara, entonces una falta se presenta en el sistema y t_i no se habilita de nuevo, donde $t_i \in T_f^{PR}$

Caso 1. Cuando t_f es una falta permanente, t_i no se habilita porque p_i perdió la marca al dispararse la transición de falta t_f. Debido a que N_f es (Q^N, M_0^N) entonces $\forall t_i \in T_f^{PR}$ y $\forall t_x \in T^N$, $D_H(t_i, t_x) < \infty$. Esto significa que una transición $t_i \in T_f^{PR}$ se intentará disparar eventualmente, justo después de activarse la señal $\lambda(t_i)$, lo cual por hipótesis es diferente de ε. De cualquier forma, t_i no puede dispararse debido a que no está habilitada, esto es, su disparo no se detecta (por hipótesis t_i es evento-detectable) y la falta t_f es detectada y aislada.

Caso 2. Cuando t_f es una falta operacional, entonces $t_i \in T_f^{PR}$ está habilitada con el disparo de t_f que marca un lugar $p_j \in P_f^{PR}$ Debido $\forall t_x \in T^N$, $D_H(t_i, t_x) < \infty$, t_i será eventualmente disparada, sin embargo, si no existe una transición de la secuencia $\sigma \in T_f^{Path}$ que haya sido disparada, entonces la única posibilidad de disparar t_i sin disparar una transición de la secuencia σ es que la transición t_f se haya disparado, entonces es en este caso cuando la falta t_f es detectada y aislada. Debido a que lo anterior se cumple para todas las transiciones de falta entonces (Q, M_0) es diagnosticable entrada-salida.

Ejemplo 2.12. Considere la *RPI* de la figura 2.8, donde existen dos faltas permanentes $T^{PF} = \{t^1_{pf}, t^2_{pf}\}$. En esta *RPI*, la transición t^1_{pf} es no diagnosticable debido a que el área de influencia generada por el *P-semiflujo* $Y_1 = [1\ 0\ 1\ 1\ 0\ 0\ 0]^T$ no comparte transiciones con el *T-semiflujo* $X_1 = [0\ 0\ 0\ 0\ 0\ 1\ 1]^T$ del comportamiento normal. Lo anterior indica que cuando una transición de falta permanente ocurre y t^1_{pf} se dispara, entonces el *T-semiflujo* X_1 puede ser disparado infinitamente sin que se detecte en un número finito de pasos el disparo de la transición t^1_{pf}. Por otro lado, la transición t^2_{pf} es diagnosticable debido a que cada *T-semiflujo* del área de influencia generada por t^2_{pf} se relaciona con todas las transiciones de la red del comportamiento normal. Así, cuando el lugar p_7 se desmarca debido al disparo de la transición de falta t^2_{pf}, el *T-semiflujo* X_1 no se disparará de manera infinita, porque la transición t_6 para ser disparada requiere la marca que se retuvo en el lugar p_7 y al no poderse disparar indica que una falta ocurrió.

2.5. Diagnosticador Centralizado en línea

El diagnosticador debe detectar y aislar las faltas usando la información de entrada-salida del sistema. El esquema de diagnosticador propuesto es una variante del que se encuentra en [Ramírez.-Treviño, et al., 2007]. El diagnosticador (ver figura 2.10) consta de cuatro módulos: el modelo diagnosticador, el cálculo del error, un algoritmo de localización de la falta y el módulo de reconstrucción de eventos. El modelo diagnosticador se construye a partir del comportamiento

normal del sistema modelado y sólo incluye las transiciones evento-detectables y lugares medibles y de riesgo del sistema. El cálculo del error obtiene la diferencia entre las salidas generadas por el sistema y el diagnosticador, cuando exista una diferencia diferente de cero, entonces se dice que un error se ha presentado en el sistema. El algoritmo de localización de la falta es una función que depende del valor del error para determinar cual es el componente del sistema que está averiado. Finalmente, el módulo reconstructor de eventos usa la información de entrada y salida del sistema para determinar que evento sucedió, el diagnosticador usa esta información y evoluciona su estado en el modelo diagnosticador.

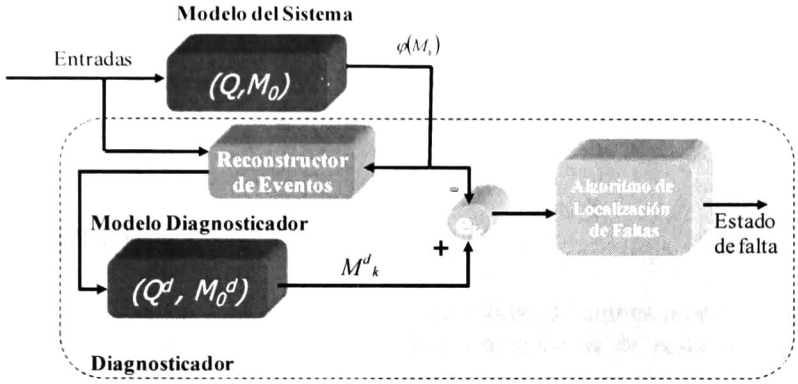

Fig. 2.10. Esquema Diagnosticador en línea

2.5.1. Modelo Diagnosticador

Definición 2.35. Sea (Q,M_0) una *RPI* representada por la ecuación de estado:

$$M_{k+1} = M_k + Cv_k$$
$$y_k = \varphi M_k$$

Donde φ es la matriz de salida de (Q,M_0) y C es la matriz de incidencia de (Q,M_0), con faltas agregadas como en la sección de modelado (ver sección 2.2). Un diagnosticador para esta red se puede construir de la siguiente forma:

$$C^d = C^N\big|_{Nd}$$

Donde C^N es la matriz de incidencia restringida al comportamiento normal, considerando sólo a un conjunto de nodos: $N_d = P_d \cup T_d$, $P_d \subseteq P$ y $T_d \subseteq T$, donde $T_d = T_f^R \cup T_f^{PR} \cup T_f^{PATH}$ representan las transiciones que son parte de la proposición 2.34 y P_d son los lugares medibles y de riesgo para detectar el conjunto de transiciones T_d, entonces (Q^d,M_0^d) es una subred de (Q,M_0) considerando sólo los nodos del comportamiento normal que sean de interés, esto es, aquellos que permitan distinguir el disparo de las faltas modeladas.

El marcado inicial del modelo diagnosticador se calcula como:

$$M_0^d = M_0\big|_{Pd}$$

Agregar las funciones λ_d y φ_d.

- $\lambda_d: T_d \rightarrow \Sigma \cup \{\varepsilon\}$, tal que, $\forall t_j, t_k \in T_d, j \neq k$, debido a que $t_j, t_k \in T$, entonces $\lambda(t_j) = \lambda(t_j)$ y $\lambda(t_k) = \lambda(t_k)$.

- φ_d: $R(Q^d,M_0^d) \rightarrow (\mathbb{Z}^+)^q \mid_{Pd}$ es una función de salida, que asocia a cada marcado en $R(C^d,M_0^d)$ un vector de salida. Aquí q es el número de salidas restringida a los lugares medibles de P_d.

Las reglas de disparo del modelo diagnosticador son las siguientes:

- Si una transición evento-detectable $t_i \in T_d$ se dispara en (Q,M_0), entonces se dispara en (Q^d,M_0^d). Esta regla puede ser implementada, debido a que cada transición t_i del modelo (Q^d,M_0^d) es evento-detectable, entonces la información de entrada-salida del sistema se puede usar para determinar el disparo de tal transición.
- Si una transición post-riesgo $t_j \in T^R$ se habilita en (Q^d,M_0^d) y el símbolo $\lambda(t_j)$ es activado en (Q,M_0), entonces t_j debe ser disparada en (Q^d,M_0^d).

2.5.2. Cálculo del Error

El error se identifica mediante la diferencia de salidas de ambos modelos (del sistema y del diagnosticador). A la salida generada por el diagnosticador se le resta la salida del sistema restringida al conjunto de lugares del modelo diagnosticador, esto es, $e_k = \varphi_d M_k^d \quad \varphi M_k|_{Pd}$, cuando la diferencia es diferente de cero significa que un error se ha detectado.

2.5.3. Detección de la falta

Una vez que se ha obtenido que $e_k \neq 0$, se verifica que si $\varphi_d C^d(\bullet,i) = - e_k$ entonces $^{\bullet\bullet}t_i \in T^{OF}$ es una falta operacional que ocurrió en el sistema, pero si $\varphi_d C^d(\bullet,i) = e_k$ entonces $^{\bullet\bullet}t_i \in T^{PF}$ es una falta permanente la que se presentó.

Ejemplo 2.13. Considera la celda de producción modelada en la figura 2.7. Debido a que la red cumple las condiciones de la proposición 2.34 entonces es diagnosticable entrada-salida y el diagnosticador puede ser construido a partir de este modelo. El modelo diagnosticador se visualiza en la figura 2.11.

La matriz $\varphi_d C^d$ es la siguiente:

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

Si la secuencia de transiciones $\sigma = t_1 t_{of1}$ se ejecuta en el sistema entonces el lugar marcado en el sistema es p_8. Para poder detectar la falta operacional en el diagnosticador se ejecuta la transición t_1 y se queda con marca p_5. El sistema evoluciona disparándose en un número finito de

pasos la transición t_7, el módulo de reconstrucción de eventos detecta la transición y cuando se quiere disparar en el diagnosticador no se puede debido a que el lugar p_8 no se encuentra marcado, entonces $e_k = \varphi_d M_k^d - \varphi M_k$, $e_k = [0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T - [1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T = [-1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$, se verifica que si $\varphi_d C^d(\bullet,i) = -e_k$ el cual representa que t_7 no fue disparada en el diagnosticador y por lo tanto la falta operacional t_{of1} se detecta y se localiza.

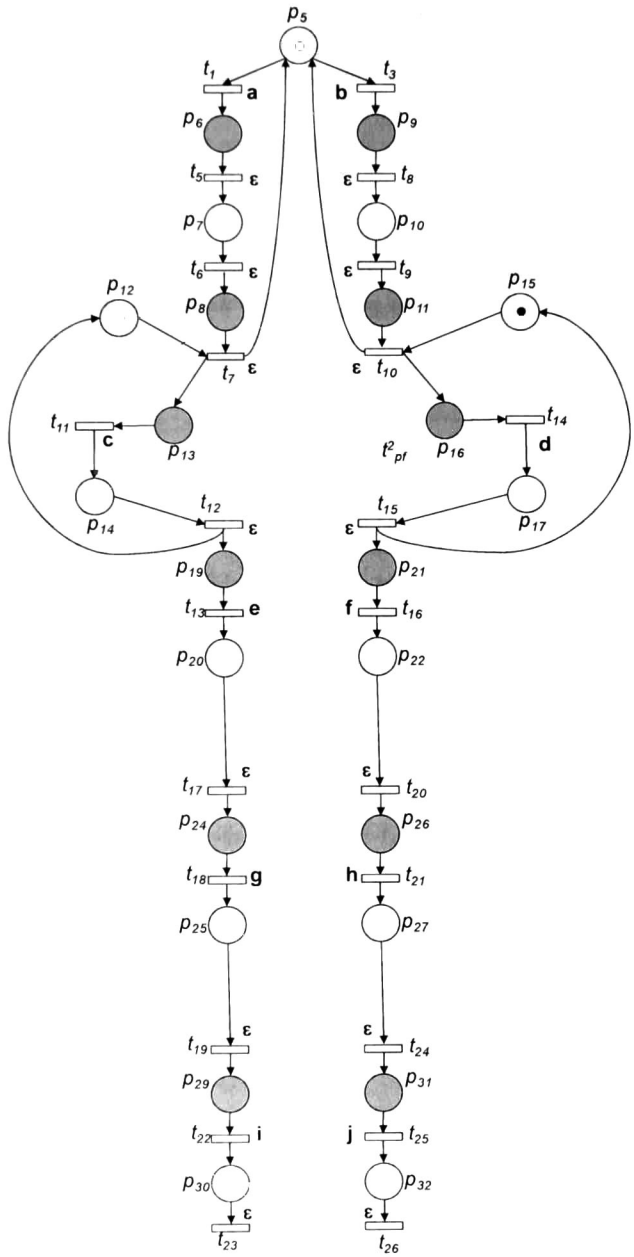

Fig. 2.11. Modelo diagnosticador

Capítulo 3 Diagnóstico Distribuido de Sistemas de Eventos Discretos

Resumen: *En este capítulo se presenta el uso de diagnosticadores distribuidos para detección y localización de las faltas de un modelo distribuido. El modelo distribuido se obtiene a partir de un modelo global que cumple con la propiedad de diagnosticabilidad centralizada. Se adapta la propiedad de diagnosticabilidad con respecto a una distribución. Se definen dos tipos de diagnosticabilidad: a) local: cuando un módulo puede detectar la falta de manera local, es decir, con la información de entrada-salida que pertenece a la estructura del módulo, o bien, b) condicional: un módulo incorpora salidas asociadas a otros módulos del sistema, las salidas agregadas permiten detectar el disparo de transiciones locales. Se utiliza un diagnosticador por cada módulo para el proceso de detección y localización de las faltas modeladas en cada módulo. Finalmente se presenta la implementación de diagnosticadores redundantes para agregarle confiabilidad al sistema de diagnóstico distribuido en línea.*

3.1. Modelo Distribuido

En el capítulo anterior se explicó la caracterización de la propiedad de diagnosticabilidad en modelos centralizados y la utilización de un diagnosticador centralizado para realizar diagnóstico en línea, sin embargo, algunas de las desventajas de utilizar un diagnosticador centralizado son: a) alta complejidad espacial – el espacio de estados del modelo diagnosticador para sistemas modelados con autómatas finitos usualmente es el producto de los conjuntos de estados de los componentes, o bien, la utilización de modelos complejos con poca claridad de interpretación, b) poco robusto – si el diagnosticador centralizado falla, la función de diagnóstico del sistema no se realiza, en un ambiente distribuido existen varios diagnosticadores que se encargan de monitorear al sistema, si alguno falla, el sistema queda parcialmente diagnosticado por otros, c) cambio de modelos – cualquier cambio en el sistema, esto es, agregar nuevos componentes, remover componentes o cambiar algunas conexiones de entrada-salida, puede forzar la construcción de un nuevo modelo para el diagnosticador. Para manejar algunos de los aspectos negativos del diagnosticador centralizado se propone construir un diagnosticador distribuido, aprovechando la ventaja de trabajar con sistemas grandes y complejos. Cada diagnosticador que pertenece al esquema distribuido se encarga de detectar y localizar las faltas en submodelos del sistema global, el conjunto total de submodelos del diagnosticador distribuido forma una distribución del modelo global $RPI(Q,M_0)$. El conjunto de submodelos puede tener nodos comunes, los cuales representan la comunicación en el diagnosticador distribuido. Se considera que el modelo $RPI(Q,M_0)$ cumple con las condiciones de la proposición 2.34.

Definición 3.1. Sea (Q,M_0) una *RPI*. Un módulo $\mu_k = (N_k, \Sigma_k, \lambda_k, \varphi_k)$ de la *RPI* es una subred formada por lugares y transiciones del modelo global (Q,M_0), donde:

$N_k = (T_k, P_k, I_k, O_k, I_k^C \; O_k^C \; M_{0k})$.

$T_k \subseteq T$.

$P_k = P_k^L \cup P_k^C$; $P_k^L \subseteq P$; P_k^C representa los lugares de comunicación entre módulos y son copia de un lugar P_k^L que pertenece algún otro módulo $l \neq k$. P_k^C es el conjunto de lugares que permiten que las transiciones del módulo sean evento-detectables. $M(P_k^C) = M(P_l^L)$. P_k^C puede ser vacío.

$I_k(O_k)$: $P_k^L \times T_k \rightarrow Z^+$, tal que, $I_k(p_i,t_j) = I(p_i,t_j)$ $(O_k(p_i,t_j) = O(p_i,t_j))$, $\forall p_i \in P_k^L$ *and* $\forall t_j \in T_k$.
$I_k^C(O_k^C)$: $P_k^C \times T_l \rightarrow Z^+$, tal que, $I_k^C(p_i,t_j) = I(p_i,t_j)$ $(O_k^C(p_i,t_j) = O(p_i,t_j))$, $\forall p_i \in P_k^C$ and $\forall t_j \in T_l$, $l \neq k$. $I_k^C(O_k^C)$ son los arcos de entrada (salida) desde $p_i \in P_k^C$ *a otras transiciones de otros módulos.*
$M_{0k} = M_0 \mid_{Pk}$
$\Sigma_k = \{\alpha \in \Sigma \mid \exists t_i, t_i \in T_k, \lambda(t_i) = \alpha\}$
$\lambda_k : T_k \rightarrow \Sigma_k \cup \{\varepsilon\}$, t.q. $\lambda_k(t_i)$ $\lambda_k(t_i) = \lambda(t_i)$ y $t_i \in T_k$.
φ_k $(\mu_k, M_{0k}) \rightarrow (Z^+)^{qk}$, q_k está restringida a las salidas asociadas a P_k.

Definición 3.2. Sea (Q,M_0) una *RPI*. Una distribución DN_i de (Q,M_0) es un conjunto finito de m módulos, esto es, $DN_i = \{\mu_1, \mu_2, \ldots, \mu_m\}$. La distribución DN_i cumple con las siguientes condiciones:

1. $\bigcap_{k=1}^{m} P_k^L = \phi; \bigcup_{k=1}^{m} P_k^L = P$ (es una partición sobre los lugares P_k^L, sin considerar el conjunto P_k^C)
2. $\bigcup_{k=1}^{m} T_k = T$
3. $\forall t_f \in (T^{OF} \cup T^{PF})$ y $^{\bullet} t_f \in P_k$, entonces $t_f^{\bullet} \in P_k$

El conjunto de lugares de comunicación $P'_{com} = \bigcup_{i=1}^{m} P_k^C$ de una distribución DN_i representa las salidas duplicadas y asociadas a los lugares de un módulo μ_k para preservar la evento-detectabilidad de $T_k \in \mu_k$.

Definición 3.3. Sea DN_i una distribución de (Q,M_0) y $\sigma = t_i t_j \ldots t_k \ldots$ una secuencia de transiciones de $\pounds(Q,M_0)$. La proyección del lenguaje del sistema $\pounds(Q,M_0)$, sobre los módulos de una distribución $\mu_k \in DN_i$, está dado por la siguiente proyección PT_k:

$$PT_k: \pounds(Q,M_0) \rightarrow \pounds(\mu_k,M_{0k})$$
$$\forall t_i t_j \ldots t_s t_q \in \pounds(Q,M_0)$$
$$PT_k(\varepsilon) = \varepsilon$$
$$PT_k(t_i t_j \ldots t_s t_q) = \begin{cases} PT_k(t_i t_j \ldots t_s) & si\ t_q \notin T_k \\ PT_k(t_i t_j \ldots t_s) t_q & si\ t_q \in T_k \end{cases}$$

Existe la proyección de símbolos de entrada-salida sobre los símbolos de entrada-salida de los módulos. Sea $\omega = (\alpha_0,y_0)(\alpha_1,y_1)\ldots(\alpha_n,y_n)$ una secuencia de símbolos de (Q,M_0):

$P\Lambda_{\mu k}(\omega) = ((P_{INk}\alpha_0, P_{OUTk}y_0),(P_{INk}\alpha_1, P_{OUTk}y_1)\ldots(P_{INk}\alpha_n, P_{OUTk}y_n))$ donde:

$$P_{INk}(\alpha_i) = \begin{cases} \varepsilon & si\ \alpha_i \notin \sum_k \\ \alpha_i & si\ \alpha_i \in \sum_k \end{cases} \quad \text{y}$$

$$P_{OUTk}\left(y_i = \begin{bmatrix} y_i(1) \\ \vdots \\ y_i(q) \end{bmatrix}\right) = \begin{bmatrix} y_i'(1) \\ \vdots \\ y_i'(q) \end{bmatrix}$$

donde $y_i(s) = 0$ si el lugar medible no pertenece a μ_k

donde $y_i(s) = y_i(s)$ de otra forma.

Definición 3.4. El conjunto de transiciones frontera de un módulo μ_k es $T^k_{borde} = \{t_i \mid t_i \in \{T_k \cap T_z\}\}$ y denota las transiciones comunes que son disparadas al menos por otro módulo μ_z, tal que, μ_k y $\mu_z \in DN_i$.

Ejemplo 3.1. Considere la figura 3.1, que representa una distribución del modelo global presentado en la figura 2.7. La distribución está formada por tres módulos: $DN_1 = \{\mu_1, \mu_2, \mu_3\}$, donde el

módulo 1 necesita dos lugares del módulo 2 para preservar la evento-detectabilidad de transiciones frontera: $P_1^C = \{p_{12}, p_{15}\}$ y el módulo 3 necesita dos salidas asociadas al módulo 2: $P_3^C = \{p_{20}, p_{22}\}$, por lo tanto el total de lugares de comunicación para la distribución es: $P^1_{com} = \{ p_{12}, p_{15}, p_{20}, p_{22}\}$. El conjunto de transiciones frontera para cada módulo es: $T^1_{borde} = \{t_7, t_{10}\}$, $T^2_{borde} = \{t_7, t_{10}, t_{17}, t_{20}\}$ y $T^3_{borde} = \{t_{17}, t_{20}\}$. Si en el sistema se da la secuencia de transiciones siguiente: $\sigma = t_1t_5t_6t_7t_{11}t_{12}t_{13}t_{17}$ al proyectarse sobre el módulo 1 y el módulo 2 se obtiene: $PT_1 = t_1t_5t_6t_7$ y $PT_2 = t_7t_{11}t_{12}t_{13}t_{17}$ respectivamente.

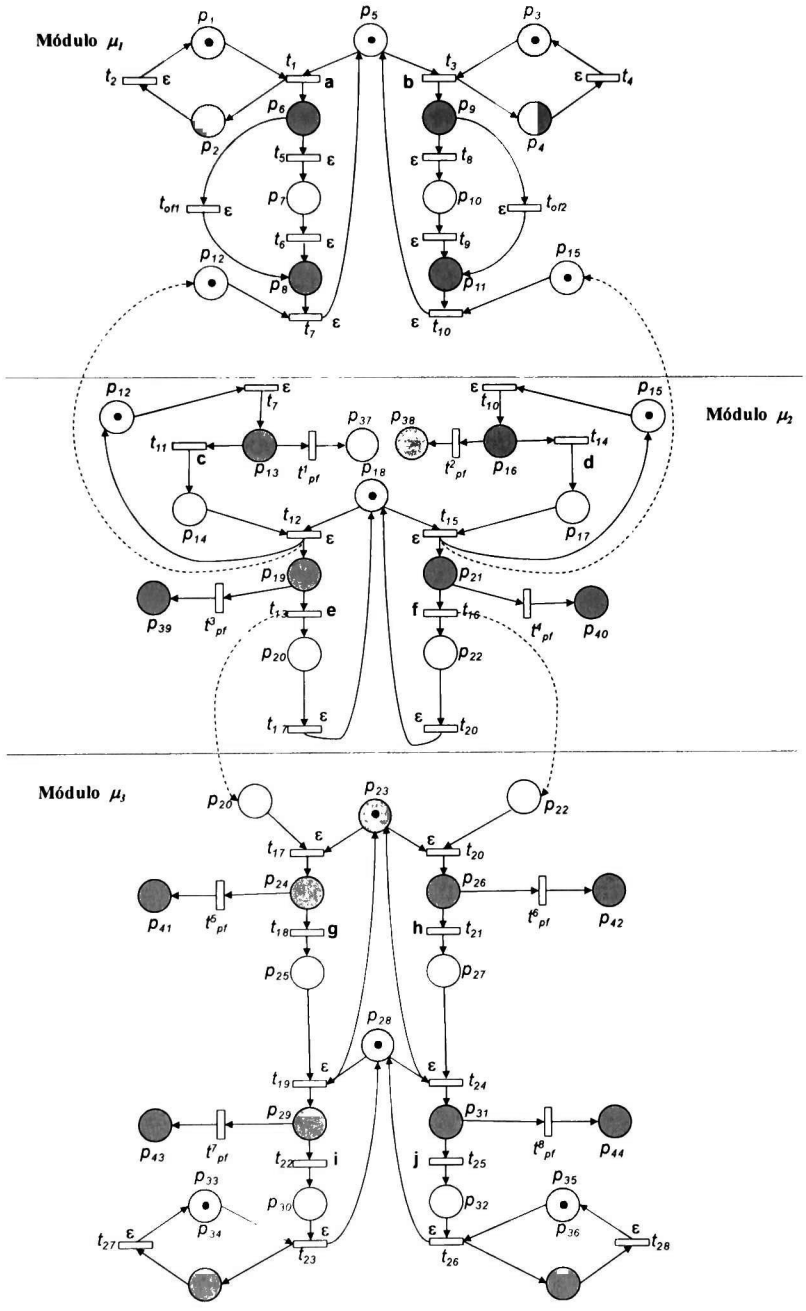

Fig. 3.1. Modelo Distribuido de la celda de producción

3.2. *Diagnosticabilidad entrada-salida distribuida*

En esta sección se muestra que la diagnosticabilidad adaptada a un enfoque distribuido es equivalente a la centralizada.

El conjunto de nodos de un módulo $\mu_x \in DN_i$ se particiona en el conjunto de nodos locales de falta ($P^{PF}{}_x$), lugares codificando estados locales de falta permanente, y ($T^{PF}{}_x \cup T^{OF}{}_x$) transiciones locales conduciendo a lugares de falta permanente y transiciones de falta operacional local, y de igual forma el conjunto de nodos de funcionamiento normal ($P^N{}_x$ y $T^N{}_x$); de esta manera, $P_x = P^F{}_x \cup P^N{}_x$ y $T_x = T^F{}_x \cup T^{OF} \cup T^N{}_x$.

La adaptación de la caracterización de diagnosticabilidad centralizada a una distribución DN_i diagnosticable entrada-salida se basa en distinguir del conjunto de alcanzabilidad de un módulo local $R(\mu_x, M_{0x})$ los marcados normales y de falta locales, donde todos los marcados de falta locales deben ser distinguibles de aquellos marcados que representan el funcionamiento normal como se explicó en la sección 2.3. $R(\mu_x, M_{0x})$ es particionado en los siguientes dos subconjuntos: a) el conjunto de marcados de falta $LF = \{M_{kx} \mid \exists p_k \in P^F{}_x\}$, tal que $M_{kx}(p_k) > 0$, $M_{kx} \in R(\mu_x, M_{0x})\}$ y, b) el conjunto de estados normales locales $LN = R(\mu_x, M_{0x}) - LF$

Ejemplo 3.2. Considere la *RPI* de la figura 3.2, donde existe un modelo global que tiene dos faltas permanentes $T^{PF} = \{t^1_{pf}, t^2_{pf}\}$ y una falta operacional $T^{OF} = \{t_{of1}\}$, además se presenta una distribución compuesta por dos módulos. Los correspondientes grafos de alcanzabilidad de cada uno de los módulos presentados en la figura 3.3, muestran que tanto después de la presencia de las faltas $t^1_{pf} \in \mu_1$ y $t^2_{pf} \in \mu_2$, será inevitable un bloqueo en cada uno de los módulos. Mientras que para la falta operacional t_{of1} será distinguible cuando no se disparé la transición t_3 y el sistema evolucione con el disparo de t_1.

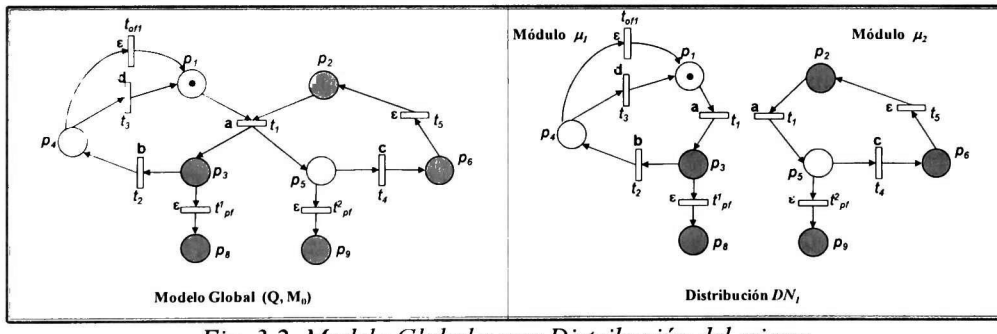

Fig. 3.2. Modelo Global y una Distribución del mismo

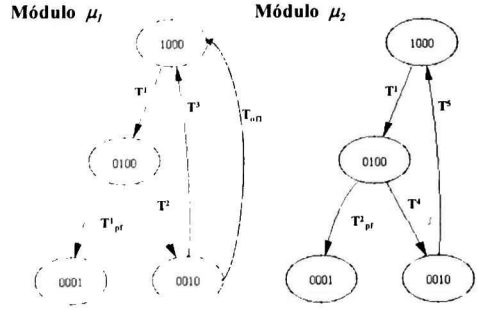

Fig. 3.3. Grafo de alcanzabilidad de la distribución de la figura 3.2

Definición 3.5. El conjunto de todas las secuencias de entrada-salida que marcan un lugar $p_i \in P^i_{com}$ en un módulo μ_x está indicado por: $\Lambda^{int}{}_x(\mu_x, M_{0x}) = \{\omega_x \in \Lambda^{int}{}_x(\mu_x, M_{0x}) | \exists\sigma_x$ tal que $M_{0x} \xrightarrow{\sigma_x} M_{jx}$ y M_{jx} marca el lugar $p_i\}$.

Definición 3.6. Sea $M_z \in R(Q, M_0)$ un marcado del sistema, y sea $DN_i = \{\mu_1, \mu_2, \ldots, \mu_m\}$ una distribución con m módulos, donde $\forall\mu_k \in DN_i$ se dispara una secuencia local de transiciones σ_k, el disparo alcanza un marcado local M_{ik} en los módulos, esto es, $M_{0k} \xrightarrow{\sigma_k} M_{ik}$, entonces, $\bigcup_{k=1}^{m} M_{ik} = M_z$, cada M_{ik} se representa como un vector de 0s y 1s (se trabaja con redes de Petri binarias), cada posición del vector implica un lugar del módulo, esto es, $M_{ik} = [p_1\ p_2\ \ldots\ p_x]^T$ La operación de unión $\cup$ se aplica de la siguiente forma:

a) $0 \cup 0 = 0$,
b) $0 \cup 1 = 1$,
c) $1 \cup 1 = 1$, cuando $\exists p_i \in P_K$, t.q., $p_i \in P^i_{com}$.

En esta tesis se definen dos tipos de diagnosticabilidad basada en el conjunto de nodos de cada módulo: 1) Diagnosticabilidad local y, 2) Diagnosticabilidad condicional.

Definición 3.7. (Diagnosticabilidad Local). Un módulo $\mu_x \in DN_i$ es diagnosticable localmente en $k < \infty$ pasos si cualquier marcado $M_{fx} \in LF$ es distinguible de cualquier otro marcado $M_{kx} \in R(\mu_x, M_{0x})$ usando símbolos de entrada-salida locales $\omega_x \in \Lambda^k{}_x(\mu_x, M_{0x}) \cup \Lambda_{Bx}(\mu_x, M_{0x})$.

Definición 3.8. (Diagnosticabilidad Condicional). Un módulo $\mu_x \in DN_i$ es diagnosticable de manera condicional en $k < \infty$ pasos si cualquier marcado $M_{fx} \in LF$ es distinguible de cualquier otro $M_{kx} \in R(\mu_x, M_{0x})$ usando los símbolos de entrada-salida locales $\omega_x \in \Lambda^k{}_x(\mu_x, M_{0x}) \cup \Lambda_{Bx}(\mu_x, M_{0x})$ y además usando las secuencias de símbolos $\omega_z \in \Lambda^{int}{}_z(\mu_z, M_{0z})$ de otro módulo.

Las definiciones permiten distinguir que un módulo es localmente diagnosticable, si cada falta modelada es posible detectarla sólo con la información local, de lo contrario, el módulo es diagnosticable condicionalmente. En la figura 3.1 se observa que los módulos 1 y 3 son condicionalmente diagnosticables, mientras que el módulo 2 es localmente diagnosticable. La diagnosticabilidad condicional de un módulo $\mu_x \in DN_i$ considera el envío de mensajes entre módulos representado por los lugares P^i_{com}, los cuales deben ser lugares medibles.

Proposición 3.9. Sea (Q,M_0) una *RPI* y DN_i una distribución que cumple con la definición 3.2. Si (Q,M_0) es diagnosticable entrada-salida que cumple con la proposición 2.34, entonces DN_i es diagnosticable entrada-salida de manera distribuida.

Demostración. Se supone que (Q,M_0) es diagnosticable entrada-salida. Si existe una secuencia finita de símbolos de entrada-salida $\omega \in \Lambda^k(Q,M_f) \cup \Lambda_B(Q,M_f)$, y $\sigma = t_i t_j t_k \ldots t_m$ es la secuencia de disparo de transiciones que genera la palabra ω, t.q., $M_0 \longrightarrow M_k$, de tal forma que se ejecuta una transición de falta en la secuencia σ, entonces el marcado alcanzado es de falta, $M_k \in F$. Por la proposición 2.34, M_k es distinguible de cualquier otro marcado $M_k \in R(Q,M_0)$ entonces (Q,M_0) es diagnosticable entrada-salida. Debido a que DN_i es una distribución de (Q,M_0), suponga que la secuencia σ se proyecta en los módulos $\mu_x \in DN_i$, esto es, $\forall\mu_x\ PT_x(\sigma)$, por lo tanto la secuencia genera los marcados locales correspondientes M_{ix} por la definición 3.3, y por definición 3.6, $\bigcup_{i=1}^{m} M_{ix} = M_k$, entonces, $\exists M_{ix} \in LF$. Sean $\sigma_1, \sigma_2, \ldots, \sigma_x$ las secuencias que son parte de $\sigma = \sigma_1\sigma_2 \ldots \sigma_x$, suponga que σ_1 se dispara en un módulo μ_j, tal que, $M_{0j} \xrightarrow{\sigma_1} M_{j1}$, σ_2 se dispara en el módulo μ_l, tal que, $M_{0l} \xrightarrow{\sigma_2} M_{l1}$

..., σ_x se dispara en μ_x, tal que, $M_{0x} \longrightarrow M_{ix}$, y σ ocurre, si la secuencia σ_1 seguida por la secuencia σ_2,... seguida por la secuencia σ_m ocurre en los módulos correspondientes. Entonces por definición 3.7 y 3.8 μ_x puede distinguir cualquier marcado $M_{ix} \in LF$, de cualquier otro $M_{jx} \in R(\mu_x, M_{0x})$, por lo que siempre existe un módulo μ_x de la distribución DN_i que puede distinguir el marcado de falta correspondiente M_{ix}; como μ_x puede ser cualquier módulo μ_x puede ser local o condicionalmente entrada-salida diagnosticable, por consiguiente, DN_i es diagnosticable entrada-salida. □

La proposición anterior considera ambos casos (módulos locales y condicionalmente diagnosticables) para establecer la propiedad de diagnosticabilidad entrada-salida de una distribución DN_i.

Definición 3.10. Sea DN_i una distribución de (Q,M_0) como la definición 3.2. DN_i es local (condicionalmente) diagnosticable entrada-salida de manera distribuida si y sólo si cada $\mu_k \in DN_i$ es local (condicionalmente) diagnosticable entrada-salida.

Ejemplo 3.3. Considere la *RPI* de la figura 3.4, donde existe un modelo global que tiene dos faltas permanentes $T^{PF} = \{t^1_{pf}, t^2_{pf}\}$ y una falta operacional $T^{OF} = \{t_{of1}\}$, además se presenta una distribución compuesta por dos módulos. El módulo μ_1 es localmente diagnosticable, mientras que el módulo μ_2 es condicionalmente diagnosticable ya que necesita la salida asociada al lugar p_1 para conocer que se ha marcado el lugar de riesgo p_5 y que probablemente una falta modelada por t^2_{pf} pueda ocurrir. El lugar p_1 pertenece al módulo μ_1, por lo tanto el módulo μ_2 es condicionalmente diagnosticable.

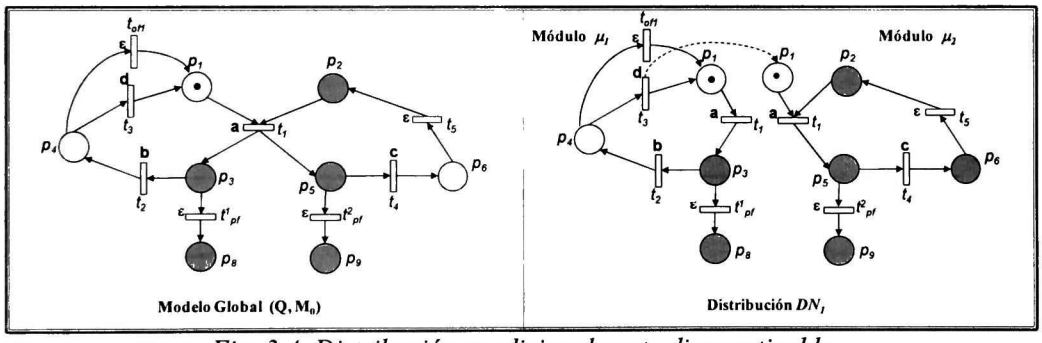

Fig. 3.4. Distribución condicionalmente diagnosticable

3.3. Diagnosticador Distribuido

Este trabajo considera el esquema para diagnóstico en línea distribuido inicialmente propuesto en [Arámburo-Lizárraga, et al., 2005] el cual se muestra en la figura 3.5. En este esquema existen m diagnosticadores locales para cada uno de los m módulos de la distribución del modelo global, esto es, $\forall \mu_k \in DN_i$ existe un diagnosticador local. La distribución se implementa en diferentes *CPU's* microprocesadores.

Si existen lugares que son de comunicación en una distribución DN_i, es decir, $P^i_{com} \neq \varnothing$ entonces, existe comunicación entre los diferentes diagnosticadores de los módulos de la distribución. Las salidas asociadas a los lugares P^i_{com} son necesarias para poder distinguir el

disparo de transiciones frontera $\bigcup_{k=1}^{m} T_{borde}^{k}$ Un diagnosticador para un módulo $\mu_x \in DN_i$ está compuesto por cuatro procedimientos:

- *Reconstructor de Eventos Locales.* Este procedimiento monitorea los símbolos de entrada-salida del sistema y detecta la transición disparada $t_x \in T_x$.

- *Procedimiento de Diagnóstico local.* Este procedimiento compara las salidas del sistema $\varphi(M_k)$ y $\varphi_x(M_{kx})$, donde $\varphi(M_k)$ se restringe a los lugares P_x que sean medibles y determina si ocurren faltas locales.

- *Modelo diagnosticador local.* Es el módulo μ_x de la distribución, donde monitorea un subconjunto de entradas-salidas del comportamiento del sistema correspondiente al módulo $\mu_x \in DN_i$. Si una transición $t_\lambda \in T$ se dispara, y $t_x \in T_k$ entonces también se dispara en el módulo μ_x y se obtiene la salida correspondiente.

- *Módulo de Comunicación.* Implementa el protocolo de comunicación entre los diagnosticadores. Existe comunicación:
  Si el procedimiento de detección local de eventos detecta salidas necesarias para otros módulos.
  Si el procedimiento de diagnóstico detecta una falta.
  - La comunicación se implementa por primitivas de comunicación como send/receive a través de la red de comunicación. En la arquitectura propuesta siempre se detecta el disparo de las transiciones involucradas en las partes críticas del sistema, debido a que DN_i es una distribución entrada-salida diagnosticable.

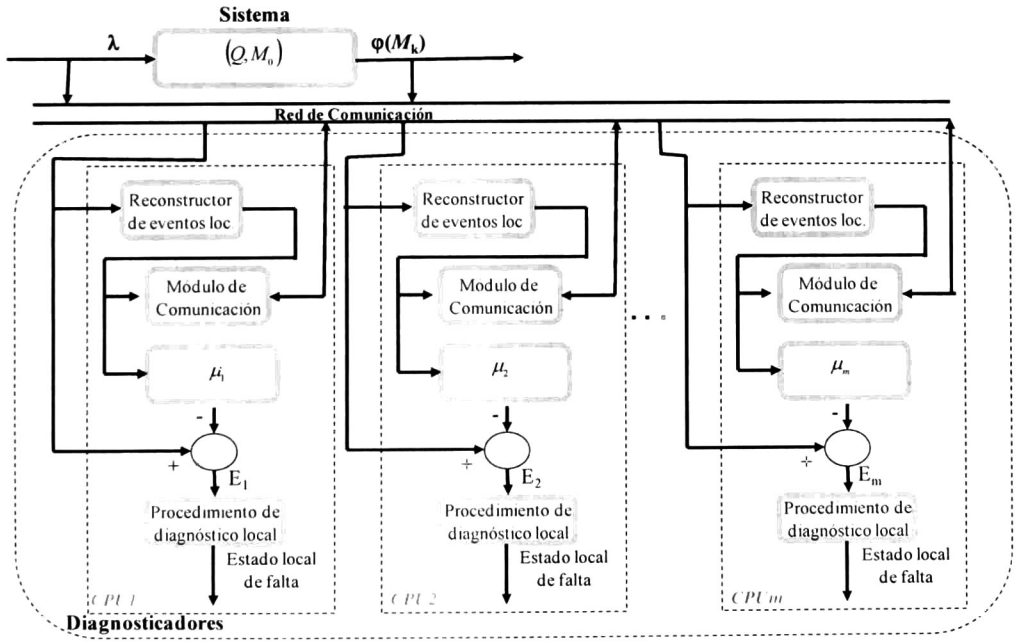

Fig. 3.5. Esquema de diagnóstico distribuido en línea

3.3.1. Modelo Diagnosticador Local

Un modelo diagnosticador para cada uno de los módulos que forman la distribución, se puede construir utilizando el modelo diagnosticador propuesto en la sección 2.5 a cada módulo.

- Para cada módulo $\mu_x \in DN_i$ obtener el conjunto de transiciones evento-detectables T_f^R, T_f^{PR} son evento detectables, y al menos una transición por cada $\sigma \in T_f^{Path}$ de cada una de las faltas f_x que son parte de μ_x.

- Agregar el siguiente conjunto de lugares: ${}^\bullet T_f^R$, ${}^\bullet T_f^{PR}$ y ${}^\bullet T_f^{Path}$, que sean lugares medibles y el conjunto $T_f^{R\bullet}$, $T_f^{PR\bullet}$ y $T_f^{Path\bullet}$

- Incorporar los símbolos de entrada asociados a las transiciones y las salidas asociadas a los lugares de los conjuntos anteriores.

Las reglas de disparo del modelo diagnosticador son las siguientes:

- Si una transición evento-detectable $t_i \in \mu_x$ se dispara en (Q,M_0), entonces se dispara en μ_x. Esta regla puede ser implementada, debido a que cada transición t_i del modelo μ_x es evento-detectable, entonces la información de entrada-salida del sistema se puede usar para determinar el disparo de tal transición.

- Si una transición post-riesgo $t_j \in T^R$ se habilita en μ_x y el símbolo $\lambda(t_j)$ es activado en (Q,M_0), entonces t_j debe ser disparada en μ_x.

Ejemplo 3.2. Considera la figura 3.1, que representa una distribución del modelo global presentado en la figura 2.7. En la figura 3.6 se puede observar el modelo de referencia para cada diagnosticador del módulo de la distribución.

3.3.2. Detección Local de Eventos

El diagnosticador utiliza el procedimiento de detección de eventos (ver algoritmo 3.1) para determinar qué transiciones se han disparado en el sistema, el diagnosticador utiliza un algoritmo que realiza los siguientes pasos:

- Observa un subconjunto de símbolos de entrada-salida del sistema correspondientes al módulo de la distribución, esto es, se realiza en el módulo μ_x la proyección del lenguaje PT_x de $£(Q,M_0)$. La proyección muestra que el diagnosticador para el módulo μ_x sólo observa el conjunto de transiciones que se encuentran representadas en el módulo. El disparo de transiciones que ocurren en el sistema es observado por cada módulo reconstructor de eventos del diagnosticador distribuido, de tal forma, que el diagnosticador sólo es capaz de monitorear al sistema de acuerdo a la partición del sistema generada por el módulo μ_x.
- Compara la salida actual con la anterior para detectar el disparo de una transición.
- El modelo del diagnosticador se actualiza disparando la transición evento-detectable $t_i \in T_x$.
- Evalúa si la transición $t_i \in T^x_{borde}$ y determina si la salida generada por el disparo de la transición es requerida por otros módulos.

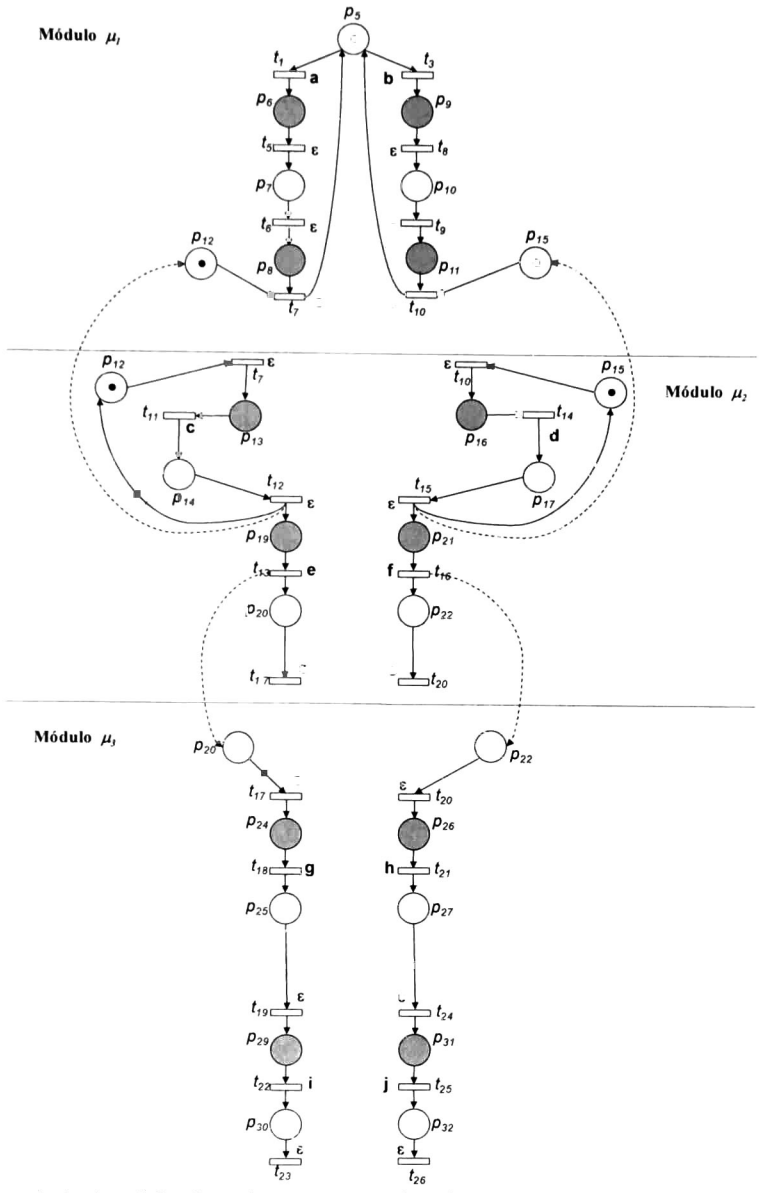

Fig. 3.6. Modelo de referencia para los diagnosticadores distribuidos

Algoritmo 3.1. Detección de Eventos Locales

Entradas: $\varphi(M_k)$, $\varphi(M_{kx})$

$\lambda(t_j)$ - el símbolo de entrada del sistema

Salidas: $(\lambda(t_j), \varphi(M_k))$

1. Constantes: $\varphi(C_x)$ - el comportamiento normal de un módulo μ_x
2. Repite

2.a. Leer $\varphi(M_k)$ y $\lambda(t_j)$

2.b. Si $\varphi(M_k) \neq \varphi(M_{k-1})$ entonces

* e = $\varphi(M_{kx}) - \varphi(M_{kx-1})$ (una columna de $\varphi(C_x)$)

* j = índice de la columna de $\varphi(C_x)$ tal que,
$\varphi(C_x)(\bullet, j) = e$, entonces t_j se disparó.

2.c. Si $t_j \in T_x$ ó $\lambda(t_j) \in \mu_x$ entonces disparar t_j en μ_x.

2.d. Realiza $E_{kx} = \varphi(M_k)|_x - \varphi(M_{kx})$

2.e. Si existen salidas de los lugares P^i_{com} en $\varphi(M_{kx})$

* Enviar un mensaje con la salida $\varphi(M_{kx})$ asociada a P^i_{com} a otros módulos

3.3.3. Detección Modular de Faltas

El error local entre la salida del sistema y la salida del modelo de un diagnosticador local correspondiente a un módulo μ_x se obtiene mediante $E_{kx} = \varphi(M_k)|_x \quad \varphi(M_{kx})$, cuando se encuentra una diferencia entre las salidas de ambos modelos, se obtiene el marcado de falta alcanzado en $\mu_x \in DN_i$. El algoritmo 3.2 se ejecuta cuando $E_{kx} \neq 0$.

Algoritmo 3.2. Detectando Marcados Locales de Falta

Entradas: $\varphi(M_{kx})$, $\lambda(t_j)$, $t_j \in P_x^R$, E_{kx}

Salidas: p_x

1.Constantes: $\varphi(C_x)$ - el comportamiento normal de un módulo μ_x

2.Repite

2.a. Leer $\varphi(M_{kx})$ y $\lambda(t_j)$

2.b. $q = \varphi(M_{kx}) \quad \varphi(M_{kx-1})$ (una columna de $\varphi(C_x)$)

2.c. i = índice de la columna de $\varphi(C_x)$, tal que,

$\varphi(C_x)(\bullet, i) = q$, entonces t_i se disparó;

2.d $^{\bullet\bullet}t_i \cap (T^{OF}{}_m \cup T^{PF}{}_m) \neq \varnothing$,

Regresa (p_x) donde $p_x \in {}^{\bullet}t_i$

Envíar el mensaje siguiente al resto de los diagnosticadores

"Una falta ocurrió en el módulo μ_x en el lugar (p_x)"

Debido a que DN_i es diagnosticable de manera distribuida entonces el paso 2.b. es ejecutado en un paso obteniendo el índice de la columna, incluso cada módulo μ_m cumple las definiciones 3.7 y 3.8, entonces el paso 2.c. también se ejecuta en un paso obteniendo el lugar de la columna que representa la transición disparada.

3.3.4. Módulo de comunicación

Los canales de comunicación entre módulos se representan en las aristas de la *RPI*, por medio de arcos punteados $I_k^C(O_k^C)$ de la definición 3.1 de distribución DN_i.

Se supone que cada módulo puede comunicarse con cualquier otro de la red. El disparo de una transición puede ser local a un módulo y puede causar cambios sólo en el diagnosticador local, ó bien involucrar comunicación con otros diagnosticadores. Sin pérdida de generalidad, se define la comunicación entre diagnosticadores de una misma distribución DN_i como sigue:

Definición 3.11. Sean μ_x y μ_y dos módulos de DN_i. Sea $P^x_{com} = \{p_i\}$. Sea *msg* el mensaje a comunicarse, donde *msg* puede ser: un mensaje que avisa que ocurrió una falta en un módulo, o bien, un mensaje para sirve para que otro diagnosticador marque el lugar de comunicación p_i en el módulo local correspondiente. La comunicación que existe entre los diagnosticadores de los módulos μ_x y μ_y, se representa a través de dos tipos de eventos: 1) *send*(μ_x, μ_y, *msg*) y *receive*(*msg*, μ_y, μ_x), donde el evento *send*(μ_x, μ_y, *msg*) consiste en el envío del mensaje *msg* del móduloμ_x al módulo μ_y y el evento *receive*(*msg*, μ_y, μ_x) consiste en la confirmación que el módulo μ_y envía al módulo μ_x de que el mensaje *msg* se ha recibido. La ejecución de los eventos *send* y *receive* se implementa cuando se detecta la ocurrencia de una falta en un módulo ó cuando una transición evento-detectable que marca al lugar p_i se dispara. En [Lampson, 1993] se proponen diferentes maneras de crear protocolos para implementación de mensajes confiables.

Definición 3.12. Sea $N = (Q, M_0)$ una *RPI* entrada-salida diagnosticable que cumple con las condiciones de la proposición 2.34 y sea DN_i una distribución para N. $\forall \mu_x \in DN_i$ el diagnosticador distribuido DD_x es una quíntupla $DD_x = (\mu_x, E_x, A, D, Com)$ donde:

- μ_x es el modelo del comportamiento normal del módulo local de la distribución.
- E_x es el conjunto de errores locales que se producen al comparar las salidas del sistema y la de módulo diagnosticador.
- D es el procedimiento de detección de eventos (algoritmo 3.1).
- A es el algoritmo de detección local de faltas que ejecuta cada módulo (algoritmo 3.2).
- *Com* puede ser cualquier protocolo de comunicación entre diagnosticadores que cumple con la definición 3.11.

El diagnosticador distribuido es la herramienta implementada de manera distribuida para el monitoreo, detección y localización de faltas modeladas en un sistema distribuido obtenido a partir de un modelo global. Los algoritmos implementados por cada módulo se encargan de monitorear al sistema en un subconjunto del mismo y el diagnosticador evoluciona conforme avanza el sistema, cuando existen módulos condicionales es necesario la comunicación entre módulos, además, es posible detectar las faltas, debido a que el modelo cumple con la proposición 2.34, por lo tanto, el diagnosticador local DD_x es capaz de detectar cuando se ha disparado una transición de falta $t_i \in (T^{OF}{}_x \cup T^{PF}{}_x)$, después del disparo de una secuencia finita de símbolos de entrada-salida que se presenta en el sistema.

3.4. Diagnosticadores Confiables

La función que desempeña un diagnosticador es la detección y localización de las faltas ocurridas en el sistema. Si el diagnosticador no desempeña esta función, la propiedad de confiabilidad (definición 1.6) del sistema no se garantiza. Confiabilidad es uno de los atributos más importantes para la tolerancia a faltas; ya que con esta propiedad se garantiza la continuidad de un servicio a pesar de faltas ocurridas. La confiabilidad está fuertemente relacionada con técnicas de redundancia tales como: duplicación o Triple Redundancia Modular (*TRM*) ver [Siewiorek y Swarz, 1992]. La redundancia utiliza componentes adicionales para detectar las faltas presentadas en el sistema a pesar de que fallen algunos componentes del mismo. En este trabajo se implementa la redundancia en los módulos del sistema del diagnosticador distribuido.

3.4.1. Esquemas Redundantes de Diagnosticadores Distribuidos

En este trabajo se propone un esquema de diagnosticador distribuido redundante compuesto por un conjunto de módulos con modelos redundantes. Se aplica redundancia en los módulos del diagnosticador distribuido para diseñar diagnosticadores confiables.

Las técnicas de Duplicación ó Triple Redundancia Modular (*TRM*) se pueden usar de la siguiente manera: se obtiene una distribución $DN_i=\{\mu_1, \mu_2,\ldots,\mu_m\}$ y ésta se distribuye sobre m computadoras aplicando duplicación ó *TRM* en el modelo diagnosticador local. En la figura 3.7 se observa *TRM* aplicada al esquema de diagnóstico en línea de la figura 3.5.

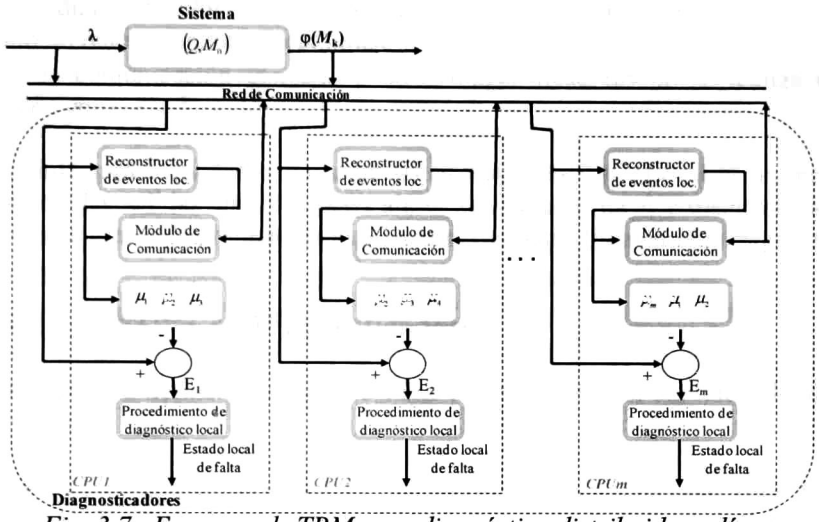

Fig. 3.7. Esquema de TRM para diagnóstico distribuido en línea

La redundancia se aplica en los modelos del sistema que tienen los diagnosticadores. La redundancia incrementa la confiabilidad en el diagnosticador de la siguiente manera: si un diagnosticador falla, existe al menos otro u otros (dependiendo de la redundancia incorporada) que continúan con la labor de diagnóstico del sistema.

Un diagnosticador con doble redundancia tiene las siguientes condiciones para decidir en conjunto con los diagnosticadores que comparte información, si ocurrió o no una falta en el sistema:

- *Si el sistema se encuentra en estado normal, el diagnosticador no detecta ninguna falta, trabaja de manera correcta.*
- *Si un diagnosticador detecta una falta en el sistema, él tiene la seguridad de que es cierto a pesar de que el otro diagnosticador no haya detectado la falta.*

La última condición sirve para identificar cuando otro diagnosticador no se encuentre trabajando correctamente.

La Triple Redundancia Modular (*TRM*) se usa como una técnica para enmascarar las faltas. En un sistema con *TRM* existe tres implementaciones del mismo sistema y su salida se incorpora a una función de mayoría en la cual se decide si una falta ha ocurrido.

Para replicar un módulo μ_x en un diagnosticador se usan las transiciones fronteras T^x_{borde} que comparten con otros módulos, y se usan aquellos que no hayan sido previamente duplicados.

Cada diagnosticador tiene información redundante del sistema para detectar faltas del sistema y faltas en el diagnosticador distribuido.

3.4.2. Tipos de Faltas en los Diagnosticadores

A continuación, se definen las faltas que suceden en los diagnosticadores distribuidos.

Definición 3.13. Las faltas que se consideran en un diagnosticador con redundancia son las siguientes:

- Caída: el diagnosticador se comporta correctamente hasta que falla y a partir de ese momento no realiza ningún procedimiento.
- Tiempo: la comunicación con uno u otros diagnosticadores no se realiza dentro de los límites de tiempo establecidos.

En la arquitectura redundante existe un sistema de votación, el cual es un mecanismo para evaluar y comparar las salidas emitidas por cada diagnosticador local y diagnosticadores vecinos redundantes y de esa forma determinar si es que ocurrió o no una falta en otro diagnosticador.

3.4.3. Detección de Faltas en Diagnosticadores

A continuación se definen los conceptos que permiten detectar las faltas del diagnosticador distribuido redundante.

Definición 3.14. V_k es una variable que el diagnosticador DD_k establece para diagnóstico del sistema. La variable V_k toma los valores de: 0 – si la falta no ocurre en el sistema, ó bien, 1- en el caso contrario.

Definición 3.15. Sea $FS_k = \{V_k, V_j, \ldots, V_z\}$ el conjunto de todas las variables de diagnóstico del sistema que crea el diagnosticador DD_k y que recibe de los diagnosticadores vecinos $DD_j \ldots DD_z$ a través del protocolo de comunicación. Dec_k es el valor que predomine sobre el conjunto FS_k y representa el valor establecido por el diagnosticador sobre el diagnóstico del sistema y $DF_k = \{DD_k, DD_j, \ldots, DD_z\}$ es el conjunto de diagnosticadores que establecen el valor de V_k diferente al valor de Dec_k y representa todos aquellos diagnosticadores vecinos que no forman parte del acuerdo generado.

Los diagnosticadores firman digitalmente los mensajes que envían, de tal forma que los resultados obtenidos por los algoritmos que tienen los diagnosticadores son originados por el correspondiente diagnosticador, no se puede hacer falsas suposiciones, con respecto a la generación de mensajes, no hay intrusos que modifiquen la fuente de donde se originan los mensajes.

El procedimiento de diagnóstico del sistema de la arquitectura redundante opera de la siguiente manera: cuando un diagnosticador detecta una falta en el sistema, espera la recepción de los mensajes que contienen la información del estado de sistema generado por otros diagnosticadores. Cada diagnosticador genera la variable V_k correspondiente al estado del sistema, la cual es utilizada por la función de mayoría, la cual devuelve el valor más frecuente entre sus argumentos. Se garantiza el acuerdo y la integridad por definición de mayoría. Cada diagnosticador aplica la función de mayoría y se genera una variable Dec_k que tiene el acuerdo alcanzado entre las variables V_k generadas por cada diagnosticador.

El diagnosticador redundante del módulo μ_x, al momento de detectar un error en el sistema $E_{kx} = \varphi(M_k)_x \quad \varphi(M_{kx})$, establece su variable de decisión V_x a estado de falta (V_x = 1), y posteriormente entra en ejecución la sección de consenso a través de la función de mayoría. La función de mayoría procesa tanto la variable de decisión local como las variables V_y, V_z emitidas por los otros diagnosticadores redundantes μ_y, μ_z y regresa el valor de estado de falta en la variable de acuerdo Dec_x. También, el diagnosticador para el módulo μ_x obtiene el conjunto DF_x para detectar los diagnosticadores vecinos que tienen el valor diferente a V_x.

Los diagnosticadores pueden fallar de manera arbitraria, en caso de que exista inconsistencia, un diagnosticador compara los mensajes que han recibido del resto para poder detectar cual es el diagnosticador inconsistente.

En [Fischer, et.al, 1985] demostraron que ningún algoritmo puede garantizar que se encuentre un consenso en un sistema asíncrono. En un sistema asíncrono, los procesos pueden responder a los mensajes en tiempos arbitrarios y en consecuencia, no se puede distinguir un proceso lento de uno que se ha caído.

Una aproximación para trabajar a pesar del resultado demostrado es considerar a los sistemas como parcialmente síncronos.

En este trabajo se considera que se distingue una falla en la comunicación entre diagnosticadores cuando un diagnosticador no ha respondido en un margen establecido de tiempo, en la práctica se transforma un sistema asíncrono en uno síncrono.

Las ventajas que se encuentran al utilizar una arquitectura redundante son las siguientes:

- El incremento de la confiabilidad de diagnóstico del sistema a pesar de que uno o más diagnosticadores fallen.
- Independencia entre diagnosticadores en la comunicación de detección de eventos comunes.
- Análisis de problemas de sistemas distribuidos como consenso, e iniciar otro tipo de problemas como el acuerdo bizantino, estados globales consistentes, entre otros.
- Los módulos son localmente diagnosticables.

Entre las desventajas se encuentran:

- El costo total de implementar la arquitectura redundante.
- Tiempos de respuesta durante la transmisión de mensajes entre los diagnosticadores.
- Tamaño de los mensajes enviados generan más tráfico en la red, por lo que existe alto costo de comunicación.

Capítulo 4 Obtención de Diagnosticador Distribuido Óptimo

Resumen: *En este capítulo se presenta el problema de distribuir el diagnosticador centralizado RPI de faltas en m CPUs, de tal forma que el número de mensajes comunicados a través de los m CPUs es el mínimo. Los mensajes de comunicación de los m diagnosticadores distribuidos sirven para poder preservar la diagnosticabilidad de las faltas. En este capítulo se presentan dos contribuciones principales: la primera contribución es la definición del grafo de comunicación dependiente entre faltas (GCDF). Los vértices de GCDF representan las señales asociadas a sensores mientras que las etiquetas de las aristas representan las faltas potenciales del sistema. Una arista $\{v_i, v_j\}$ se agrega a GCDF y se etiqueta con la falta f_k cuando f_k puede ser detectada usando la información de las señales representadas por los vértices v_i y v_j. Si dos faltas fi y fj tienen un sensor en común s_l para el proceso de detección y se encuentran en diferentes diagnosticadores, d_i y d_j, entonces se envía un mensaje con el valor del sensor común s_l de d_i a d_j. La segunda contribución es un algoritmo que explota la valencia o grado del vértice (esto es, el número de posible mensajes) en el grafo GCDF para realizar una partición m de faltas de tal forma que la cantidad de mensajes de comunicación entre los diagnosticadores sea mínimo.*

4.1. Planteamiento del problema

El diagnosticador centralizado cuenta con un modelo diagnosticable de *RPI* que cumple con la proposición 2.34, y en el capítulo 3 se define el concepto de distribución y módulos para abarcar sistemas distribuidos, y se adapta la propiedad de diagnosticabilidad al concepto de distribución, considerando módulos local y condicionalmente diagnosticables.

La propiedad de diagnosticabilidad permite detectar y localizar las faltas que se presentan en un sistema. La herramienta diagnosticador sirve para detectar y localizar las faltas en línea. Existen en la literatura 2 tipos de diagnosticadores que se usan ampliamente: diagnosticadores centralizados y distribuidos.

En [Sampath, et al., 1995] se propone una técnica para construir un diagnosticador basado en un autómata finito (*AF*). Se analiza en el diagnosticador si existen ciclos de secuencias de símbolos de entrada-salida que sean confusos, de tal forma que si se presentan ciclos con secuencias de símbolos que representan falta y también estados normales entonces no es posible distinguir si el sistema se encuentra en estado normal o de falta, por lo tanto el Sistema de Evento Discreto (*SED*) modelado en el *AF* no es diagnosticable. En este trabajo, la construcción del diagnosticador tiene complejidad exponencial en el número de estados del sistema. En [Hashtrudi, et al., 2003] se propone una extensión al trabajo previo, el diagnosticador es una máquina de estados finito, el cual toma las observaciones del sistema como entrada y genera como salida un estimado si la falta ocurre, sin embargo, el diagnosticador y el sistema inician su operación en condiciones iniciales diferentes.

Los diagnosticadores centralizados se implementan en un *CPU* por lo que se construyen modelos diagnosticadores grandes y complejos. Actualmente, el diagnosticador se implementa en sistemas distribuidos, los cuales se usan ampliamente para reducir la complejidad del sistema. Para tomar ventaja de las arquitecturas distribuidas se proponen diagnosticadores distribuidos y descentralizados.

En [Da Silveria, et al., 2002] se propone una técnica para distribuir un modelo diagnosticador centralizado. La técnica se basa en la descomposición de redes de Petri (*RP*) usando la teoría de *P*-invariantes. Sin embargo, el protocolo de comunicación usa gran cantidad de memoria y tiempo para enviar la información entre diagnosticadores, ya que cada diagnosticador envía el estado del sistema a un módulo central, es en éste último donde se decide si el sistema se encuentra en estado normal o de falta.

Basados en los resultados de [Sampath, et al., 1995], en [Debouk, et al., 2000] se propone diagnosticadores descentralizados compuestos por dos diagnosticadores. La arquitectura descentralizada está limitada a un coordinador centralizado que se encargada de realizar el diagnóstico de las faltas, además se requiere gran cantidad de memoria y tiempo para enviar toda la información al coordinador.

En [Jiroveanu y Boel, 2005] se proponen diagnosticadores distribuidos llamados agentes diagnosticadores (d-agent). Cada agente tiene una partición local del modelo basado en *RP*. Existe una partición en transiciones pero hay lugares comunes en los modelos de los agentes. La comunicación se representa mediante las marcas que son vistas en los lugares comunes. El agente monitorea al sistema utilizando una combinación de secuencias locales y secuencias marcando los lugares comunes. No se aborda la eficiencia de la comunicación pero cada agente utiliza búsqueda hacia delante y hacia atrás para obtener la representación de las secuencias del sistema y poder detectar si en la secuencia ocurrió una falta.

En [Qui y Kumar, 2005] se propone una arquitectura descentralizada basada en m diagnosticadores *AF*, donde no existe comunicación entre los diagnosticadores. Definen la propiedad de safe-codiagnosability, la cual establece que existe al menos un diagnosticador que se encarga de detectar y localizar una falta, cuando ésta ocurre en el sistema. Sin embargo, en el algoritmo para demostrar la propiedad utilizan un autómata de prueba centralizado basado en el producto síncrono del modelo de la planta, la especificación sin falta y los modelos de especificación segura.

La propuesta de este capítulo se basa en trasladar modelos centralizados que cumplen con la proposición 2.34 del capítulo 2, a una distribución DN_i que permita la construcción de m modelos diagnosticadores, donde la distribución se considera óptima porque la partición encontrada de los modelos tiene el menor conjunto de lugares P^i_{com}. La arquitectura del diagnosticador consiste de un modelo diagnosticador reducido del comportamiento normal, es decir, (Q^d, M_0^d), la obtención del error $(E_k = \varphi(M_k) - \varphi^d(M^d_k)$ donde $\varphi^d(M^d_k)$ es el vector de observación del modelo diagnosticador; un algoritmo de aislamiento de faltas y el módulo de reconstrucción de eventos.

El sistema (Q,M_0) es monitoreado por el diagnosticador centralizado en las partes críticas, esto es, donde los componentes puedan fallar. El sistema tiene un conjunto finito de distribuciones, a su vez, cada distribución puede tener m módulos posibles para diagnosticadores distribuidos.

Definición 4.1. Sea (Q,M_0) una *RPI*. El conjunto total de distribuciones posibles para (Q,M_0) se denotará: $DNT = \{DN_i, DN_j, \ldots, DN_k\}$, donde cada $DN_x \in DNT$ tiene un total de m módulos para diagnosticadores. El total de módulos por distribución está dado por: $|T^{PF}| + |T^{OF}| \geq m \geq 1$.

En *DNT* se define la relación de orden parcial $\leq$ de la siguiente manera:
$DN_j \leq DN_i$ si y sólo si $\forall \mu_x \in DN_j \ \exists \mu_y \in DN_i$, tal que, $P_k^x \subseteq P_k^y$, o bien, $\mu_x \subseteq \mu_y$, donde $|DN_i| < |DN_j|$.

El conjunto total de m módulos diagnosticadores por distribución se puede obtener mediante el número de Stirling.

- $S(F, m) = S(F, m\text{-}1) + m\, S(F\text{-}1, m)$.

donde S es el total de distribuciones, $F = |T^{PF}| + |T^{OF}|$ es el total de faltas de (Q, M_0) y m es el número de módulos de la distribución. Con este número es posible ver la complejidad del problema, ya que para modelos que tienen una cantidad pequeña de faltas, existe una cantidad muy grande de distribuciones posibles.

En la relación de orden parcial se forma una retícula donde es posible recorrer todas las distribuciones posibles de un modelo de un sistema, en este trabajo se aborda la elección de una distribución óptima con m módulos diagnosticadores, utilizando el criterio de comunicación mínima entre los diagnosticadores distribuidos. A continuación se define el planteamiento del problema de obtener un diagnosticador óptimo.

Definición 4.2. Sea (Q,M_0) una *RPI*. Sea $DN_i = \{\mu_1,\ldots,\mu_m\}$ una distribución con m módulos diagnosticadores que cumplen con la definición 3.2, y P^i_{com} el total de lugares de comunicación. Sea $DNT_m = \{DN_i, DN_j, \ldots, DN_o\}$ el total de distribuciones que tienen m módulos diagnosticadores. El problema consiste en encontrar la distribución $DN_i \in DNT_m$, tal que, el total de lugares P^i_{com} sea mínimo.

Para encontrar la distribución óptima con m módulos diagnosticadores, se realiza la obtención del conjunto de distribuciones posibles con m módulos diagnosticadores y evaluar el total de lugares medibles duplicados P^i_{com} y compararlo con todas aquellas distribuciones para obtener el valor mínimo. La complejidad de construir el conjunto de distribuciones posibles es intratable.

Ejemplo 4.1. Para ilustrar la idea del problema de obtener una distribución, en la figura 4.1 se muestra una *RPI* global y dos de sus posibles distribuciones. En la parte superior está el modelo global del sistema, mientras que en la parte inferior se encuentran dos distribuciones posibles, cada distribución tiene dos módulos. La distribución de la figura 4.1.a tiene un lugar de comunicación mientras que la distribución de la figura 4.1.b tiene 2. El problema que se está planteando en este capítulo evita la construcción de las distribuciones. El objetivo, es obtener una distribución que indique la distribución óptima con 1 lugar de comunicación como el conjunto mínimo, es decir, si es la distribución óptima, entonces obtener la distribución del sistema representado por la figura 4.1.a).

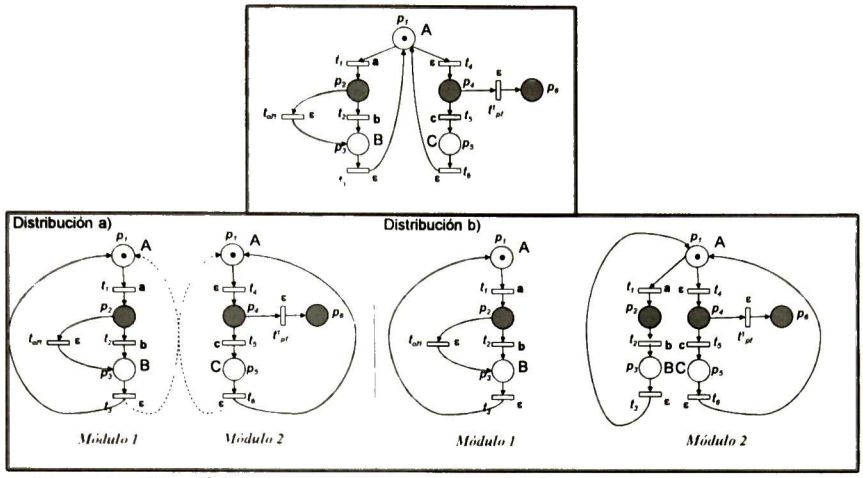

Fig. 4.1. Distribuciones posibles con 2 módulos para un modelo global

En este trabajo se propone construir la distribución utilizando la estructura de las faltas modeladas en (Q, M_0). Se crea un grafo de comunicación dependiente entre faltas *(GCDF)* donde las aristas del grafo representan a una falta del sistema, mientras que los vértices representan los sensores que utiliza cada falta para poder detectarla. A partir del grafo *GCDF* es encuentra una partición de las faltas.

4.2. Grafo de Comunicación Dependiente entre Faltas (GCDF)

En esta sección se explican una serie de definiciones y operadores que son parte del *GCDF* Las definiciones y conceptos se aplican posteriormente dentro de un algoritmo que resuelve la obtención de un diagnosticador distribuido óptimo.

Los vértices de *GCDF* representan las salidas asociados a los lugares del sistema y existe una arista e_k conectando vértices v_i, v_j, cuando existe una falta $f_z \in (T^{PF} \cup T^{OF})$ que puede ser detectada y localizada usando la información capturada por los vértices v_i, v_j. En este caso la arista e_k es etiquetada con la falta f_z. Se usa el grafo *GCDF* para encontrar la partición del conjunto de faltas en m subconjuntos, donde las faltas en cada subconjunto serán detectadas y localizadas usando un diagnosticador local. Debido a que las faltas $f_i, f_j, \ldots, f_l$ incidentes a un mismo vértice v_s, necesitan la información de salida asociada al vértice v_s, entonces si esas faltas se agrupan en q diferentes subconjuntos de falta, entonces se necesitarán q-*1* lugares de comunicación para detectar y localizar esas faltas en los diagnosticadores distribuidos.

El modelo del diagnosticador centralizado es un modelo reducido ya que se considera solamente un subconjunto de transiciones y lugares que representan las partes críticas del sistema, es decir, solo monitorea al sistema donde pueda presentarse una falta. Este subconjunto de nodos, se define como el conjunto de nodos candidatos, que permite encontrar las salidas asociadas a cada transición que permita distinguir el disparo de la secuencia, es decir, si el sistema ha evolucionado a un estado normal o de falta.

Definición 4.3. Sea $T_f^R \cup T_f^{PATH} \cup T_f^{PR}$ el conjunto de transiciones que permiten distinguir cuando se entra, se evoluciona y se sale de una zona posible de falta. El objeto soporte para estos conjuntos

de transiciones, las cuales son evento-detectables y cumplen con la condición de la proposición 2.34, se obtiene de la siguiente forma:
Si el conjunto ${}^{\bullet}T_f^{PR}$ no es medible, entonces:

- El objeto soporte para una transición $t_i \in (T_f^R \cup T_f^{PATH})$ está formado por una 3-tupla $Ot_i = <t_i, p_j, t_k>$ donde p_j es el conjunto de lugares medibles que hacen a la transición t_i evento-detectable y $t_k \in {}^{\bullet}p_j$ ó $p_j^{\bullet}$
- El objeto soporte para una transición $t_j \in T_f^{PR}$ es una tupla $Ot_j = <t_j, p_i, p_k, t_k>$ donde $p_i \in P^R$, p_k es el conjunto de lugares medibles que hacen a la transición t_j evento-detectable y $t_k \in {}^{\bullet}p_k$ ó $p_k^{\bullet}$.

De lo contrario

- El objeto soporte para una transición $t_i \in (T_f^R \cup T_f^{PR})$ está formado por una 3-tupla $Ot_i = <t_i, p_j, t_k>$ donde $p_j \in {}^{\bullet}T_f^{PR}$ es lugar de riesgo medible $t_i \in {}^{\bullet}p_j$ y $t_k \in$ ó $p_j^{\bullet}$
- El objeto soporte para una transición $t_i \in T_f^{PATH}$ está formado por una 3-tupla $Ot_i = <t_i, p_j, t_k>$ donde p_j es el conjunto de lugares medibles que hacen a la transición t_i evento-detectable y $t_k \in {}^{\bullet}p_j$ ó $p_j^{\bullet}$.

Una vez que se ha definido todos los lugares y transiciones del objeto soporte que permiten distinguir una falta del sistema, es necesario refinar este conjunto para obtener el conjunto de nodos soporte para detectar y localizar las diferentes faltas. Se usa el *P-componente* al que pertenece el lugar de riesgo donde falta puede ocurrir la falta. Se realiza una intersección con los objetos soportes y el *P-componente* respectivo, dejando solamente los lugares que pertenecen al *P-componente* si el resultado de la operación de intersección es diferente de vacío. Se analizan también si existen transiciones que comparten nodos, dejando para algunos casos los nodos del *P-componente* y para otros el objeto soporte.

Definición 4.4. Sea t_{f1} y $t_{f2} \in (T^{PF} \cup T^{OF})$ faltas en el sistema. Sea Ot_i el objeto soporte para una transición $t_i \in (T_{f1}{}^R \cup T_{f1}{}^{PATH})$ y Ot_j el objeto soporte para una transición $t_j \in T_{f1}{}^{PR}$ Sea Ot_k el objeto soporte para una transición $t_k \in (T_{f2}{}^R \cup T_{f2}{}^{PATH})$ y Ot_l el objeto soporte para una transición $t_l \in T_{f2}{}^{PR}$. Sea Y_{x1} un *P-semiflujo* de (Q, M_0) que contiene a $p_{x1} \in {}^{\bullet}t_{f1}$. Sea Y_{x2} un *P-semiflujo* de (Q, M_0) que contiene a $p_{x2} \in {}^{\bullet}t_{f2}$. El conjunto de nodos soporte para t_{f1} representa el total de nodos (lugares y transiciones) que permiten distinguir a la falta t_{f1} de otra falta t_{f2} y se define de la siguiente manera:

- $CNS(t_{f1}) = \begin{cases} Ot_i \cap Y_{x1}(p_{x1}) \cup Ot_j \cap Y_{x1}(p_{x1}) & 1 \\ Ot_i \cup Ot_j & otro\ caso \end{cases}$

1. $Si \ \cap \neq \phi \ y \forall t_p \in (Ot_i \cup Ot_j) \exists t_q \in (Ot_k \cup Ot_l), tal\ que, \lambda(t_p) = \lambda(t_q) y ({}^{\bullet}t_p = {}^{\bullet}t_q, ó, t_p = t_q^{\bullet})$

- $CNS(t_{f2}) = \begin{cases} Ot_k \cap Y_{x2}(p_{x2}) \cup Ot_l \cap Y_{x2}(p_{x2}) & 1 \\ Ot_k \cup Ot_l & otro\ caso \end{cases}$

1. $Si \ \cap \neq \phi \ y \ \forall t_p \in (Ot_i \cup Ot_j) \exists t_q \in (Ot_k \cup Ot_l), tal\ que, \lambda(t_p) = \lambda(t_q) y ({}^{\bullet}t_p = {}^{\bullet}t_q, ó, t_p = t_q^{\bullet})$

Sean t_{f1} y $t_{f2} \in (T^{PF} \cup T^{OF})$ dos transiciones de faltas del sistema, t_{f1} y t_{f2} están relacionadas si comparten lugares medibles, es decir, $t_{f1} \sim t_{f2}$ si y sólo si $CNS(t_{f1}) \cap CNS(t_{f2}) \neq \varnothing$, donde $\cap$ se restringe a lugares medibles.

Definición 4.5. El Grafo de Comunicación Dependiente entre Faltas $GCDF = (V, E)$ es un grafo donde V es el conjunto de vértices o nodos y E el conjunto de aristas. $E = \{t_f \mid t_f$ es una etiqueta que representa a la falta del sistema, $t_f \in (T^{PF} \cup T^{OF})\}$ y $V = \{p_i \cup \varepsilon \mid p_i$ son los lugares medibles que

pertenecen a $CNS(f_i)$ y ε es un vértice para indicar cuando una falta tiene asociado sólo un lugar medible}.

Una falta $t_f \in (T^{PF} \cup T^{OF})$ puede tener una cantidad diferente de lugares medibles asociados en el conjunto de nodos soporte $CNS(t_f)$, por lo tanto pueden presentarse los siguientes casos:

1. Sólo existe un lugar medible $p_i \in CNS(t_f)$, lo cual significa que $p_i \in {}^{\bullet}t_f$, entonces el lugar de riesgo es medible, por lo tanto existe una etiqueta $t_f \in E$, tal que, tiene a p_i y ε como vértices extremos a la arista representada por t_f.
2. Existen dos lugares medibles $p_i, p_j \in CNS(t_f)$, entonces existe una arista correspondiente $t_f \in E$, tal que, p_i y p_j son los vértices incidentes a la arista t_f.
3. Existen más de dos lugares medibles $p_i, p_j, \ldots, p_l \in CNS(t_f)$, entonces existen tantas aristas como el número de lugares medibles menos uno.

Dos vértices representados por p_i, p_j son adyacentes, ó vecinos, si existe una arista con la etiqueta de falta $t_f \in (T^{PF} \cup T^{OF})$, tal que, t_f es una arista que tiene a sus extremos a p_i y a p_j.

Definición 4.6. Sea $GCDF = (V,E)$ un grafo de comunicación dependiente entre faltas. El grado (o valencia) $d_G(p_i) = d(p_i)$ de un vértice $p_i \in V$, es el número de aristas con etiquetas diferentes conectadas al vértice p_i. El número $\delta(RCS) = \min \{d(p_i) \mid p_i \in V\}$ es el grado mínimo de *RCS*, mientras que el número $\Delta(RCS) = \max \{d(p_i) \mid p_i \in V\}$ es su grado máximo.

Definición 4.7. Sea (Q,M_0) una *RPI*. Sea $GCDF = (V, E)$ un grafo de comunicación dependiente entre faltas de (Q,M_0). El problema consiste en encontrar un subconjunto mínimo de vértices $C \subseteq V$, tal que C sea una partición de vértices que cubra todas las aristas de $E \in GCDF$.

La partición de vértices C indica la agrupación de las faltas en los módulos, todas las aristas conectadas a $p_i \in C$ forman un módulo para un diagnosticador, obteniendo al final un conjunto de lugares de comunicación mínimo por distribución, como lo señala la proposición siguiente.

Proposición 4.8. Sea $GCDF = (V,E)$ un grafo de comunicación dependiente entre faltas de (Q,M_0). La partición más grande tiene $m = |E|$ subconjuntos y la comunicación mínima entre los m subconjuntos está dada por: $comMin = \sum d(v_i) - |V|, \forall v_i \in V$

Demostración. Cada arista $e_k \in E$, donde $v_i, v_j \in V$ son los vértices conectados a e_k, puede ser clasificada de la siguiente manera: a) la arista (o falta) no está relacionada con ninguna otra, es decir, v_i y v_j $d(v_i) = d(v_j) = 1$, las salidas únicamente sirven para detectar la falta asociada a e_k, o bien, b) la arista e_k se relaciona con más de una arista (falta) siendo el grado de los vértices quien establece el total de faltas relacionadas.
Cuando la suma de los grados de los vértices es: $d(v_i) + d(v_j) = 2$, significa que e_k es una arista tipo a), y cuando $d(v_i) + d(v_j) > 2$ es una arista tipo b). Cada arista e_k que pertenece al primer tipo significa que la falta no está relacionada por lo tanto no comparte lugares medibles, por lo tanto, al realizar un módulo con sólo esa falta, su comunicación será 0, es decir, $comMin = 0$. Si la arista e_k pertenece al segundo tipo, entonces la suma de las valencias es mayor que 2, lo cual significa, que al menos otra arista e_j está relacionada, y que un vértice conectado v_i, v_j de la arista e_k pertenece a e_j, suponga que la arista e_k, tiene conectado a los vértices v_j, v_k, las aristas e_k y e_j comparten un vértice v_j, lo cual indica que dos faltas necesitan el sensor asociado a v_j, por lo que si se agrupan en dos subconjuntos diferentes la comunicación mínima es $comMin = 1$, ya que la valencia de $d(v_j) = 2$ y $d(v_i) = d(v_k) = 1$, la suma de los vértices es 4 menos la cantidad de vértices involucrados en las

aristas son 3, lo cual, da un total de 1 vértice que comparte más de una arista y como e_k y e_j pueden representar cualquier falta entonces, $comMin = \sum d(v_i) - |V|, \forall v_i \in V$

Ejemplo 4.2. La figura 4.2 muestra a una *RPI* y su correspondiente *GCDF*. El sistema tiene dos faltas (una operacional y una permanente). La falta operacional t_{of1} tiene los siguientes conjuntos de transiciones: $T^R_{of1} = t_1$, $T^{Path}_{of1} = t_2$, $T^{PR}_{of1} = t_3$, donde, el objeto soporte para cada transición está dado por: $Ot_1 = < t_1, p_1, t_3, t_6 >$, $Ot_2 = < t_2, p_3, t_3 >$ y $Ot_3 = < t_3, p_2, p_3, p_1, t_2, t_6 >$. El correspondiente conjunto de nodos soporte para la falta t_{of1} es: $CNS(t_{of1}) = \{t_1, p_1, t_3, t_6, t_2, p_3, t_3, p_2\}$. La falta permanente t^1_{pf} tiene los siguientes conjuntos de transiciones: $T^{R1}_{pf} = t_4$, $T^{PR1}_{pf} = t_5$, donde, el objeto soporte para cada transición está dado por: $Ot_4 = < t_4, p_1, t_3, t_6 >$ y $Ot_5 = < t_5, p_4, p_5, t_6 >$, entonces, el correspondiente conjunto de nodos soporte para la falta t^1_{pf} es: $CNS(t^1_{pf}) = \{t_4, p_1, t_3, t_6, t_5, p_4, p_5\}$. Ahora se comprueba que las faltas se encuentren relacionadas, es decir, $CNS(t_{of1}) \cap CNS(t^1_{pf}) = \{p_1\}$, por lo tanto, $t_{of1} \sim t^1_{pf}$, al estar relacionadas, el grafo *GCDF* es el indicado en la figura 4.2.

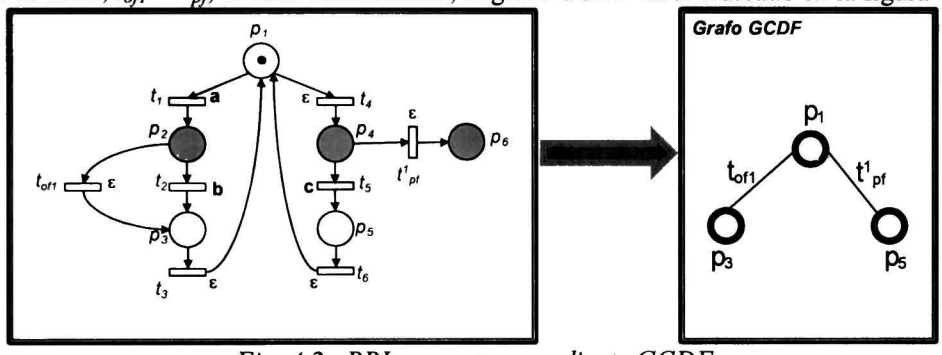

Fig. 4.2. RPI y su correspondiente GCDF

Ejemplo 4.3. La figura 4.3 muestra la *RPI* de una celda de producción (figura 2.7) y su correspondiente grafo *GCDF*. El sistema tiene 10 faltas (2 operacionales y 8 permanentes).

La falta operacional t_{of1} tiene los siguientes conjuntos:

- $T^R_{of1} = t_1$, $T^{Path}_{of1} = t_5$, $T^{PR}_{of1} = t_7$,
- $Ot_1 = < t_1, p_5, t_7, t_{10} >$, $Ot_5 = < t_5, p_7, t_6 >$ y $Ot_7 = < t_7, p_6, p_{12}, p_5, t_{12}, t_{10} >$,
- $CNS(t_{of1}) = \{t_1, p_5, t_7, t_{10}, t_5, p_7, t_6, p_6, p_{12}, t_{12}\}$.

La falta operacional t_{of2} tiene los siguientes conjuntos:

- $T^R_{of2} = t_3$, $T^{Path}_{of2} = t_8$, $T^{PR}_{of2} = t_{10}$,
- $Ot_3 = < t_3, p_5, t_7, t_{10} >$, $Ot_8 = < t_8, p_{10}, t_9 >$ y $Ot_{10} = < t_{10}, p_9, p_{15}, p_5, t_{15}, t_7 >$,
- $CNS(t_{of2}) = \{t_3, p_5, t_7, t_{10}, t_8, p_{10}, t_9, p_9, t_{15}\}$.

La falta permanente t^1_{pf} tiene los siguientes conjuntos:

- $T^{R1}_{pf} = t_7$, $T^{PR1}_{pf} = t_{11}$,
- $Ot_7 = < t_7, p_{12}, p_5, t_1, t_3, t_{12} >$ y $Ot_{11} = < t_{11}, p_{13}, p_{14}, t_{12} >$,
- $CNS(t^1_{pf}) = \{t_7, p_{12}, t_{12}, t_{11}, p_{13}, p_{14}\}$.

La falta permanente t^2_{pf} tiene los siguientes conjuntos:

- $T^{R2}_{pf} = t_{10}$, $T^{PR2}_{pf} = t_{14}$,
- $Ot_{10} = < t_{10}, p_{15}, p_5, t_1, t_3, t_{15} >$ y $Ot_{15} = < t_{14}, p_{16}, p_{17}, t_{15} >$,
- $CNS(t^2_{pf}) = \{t_{10}, p_{15}, t_{15}, t_{14}, p_{16}, p_{17}\}$.

La falta permanente t^3_{pf} tiene los siguientes conjuntos:

- $T^{R3}_{pf} = t_{12}$, $T^{PR3}_{pf} = t_{13}$,
- $Ot_{12} = < t_{12}, p_{14}, t_{11} >$ y $Ot_{13} = < t_{13}, p_{19}, p_{20}, t_{17} >$,
- $CNS(t^3_{pf}) = \{t_{12}, p_{14}, t_{11}, t_{13}, p_{19}, p_{20}, t_{17}\}$.

La falta permanente t^4_{pf} tiene los siguientes conjuntos:

- $T^{R4}_{pf} = t_{15}$, $T^{PR4}_{pf} = t_{16}$,
- $Ot_{15} = < t_{15}, p_{17}, t_{14} >$ y $Ot_{16} = < t_{16}, p_{21}, p_{22}, t_{20} >$,
- $CNS(t^4_{pf}) = \{t_{15}, p_{17}, t_{14}, t_{16}, p_{21}, p_{22}, t_{20}\}$.

La falta permanente t^5_{pf} tiene los siguientes conjuntos:

- $T^{R5}_{pf} = t_{17}$, $T^{PR5}_{pf} = t_{18}$,
- $Ot_{17} = < t_{17}, p_{20}, t_{13} >$ y $Ot_{18} = < t_{18}, p_{24}, p_{25}, t_{19} >$,
- $CNS(t^5_{pf}) = \{t_{17}, p_{20}, t_{13}, t_{18}, p_{24}, p_{25}, t_{19}\}$.

La falta permanente t^6_{pf} tiene los siguientes conjuntos:

- $T^{R6}_{pf} = t_{20}$, $T^{PR6}_{pf} = t_{21}$,
- $Ot_{20} = < t_{20}, p_{22}, t_{16} >$ y $Ot_{21} = < t_{21}, p_{26}, p_{27}, t_{24} >$,
- $CNS(t^6_{pf}) = \{t_{20}, p_{22}, t_{16}, t_{21}, p_{26}, p_{27}, t_{24}\}$.

La falta permanente t^7_{pf} tiene los siguientes conjuntos:

- $T^{R7}_{pf} = t_{19}$, $T^{PR7}_{pf} = t_{22}$,
- $Ot_{19} = < t_{19}, p_{25}, t_{18} >$ y $Ot_{22} = < t_{22}, p_{29}, p_{30}, t_{23} >$,
- $CNS(t^7_{pf}) = \{t_{19}, p_{25}, t_{18}, t_{22}, p_{29}, p_{30}, t_{23}\}$.

La falta permanente t^8_{pf} tiene los siguientes conjuntos:

- $T^{R8}_{pf} = t_{24}$, $T^{PR8}_{pf} = t_{25}$,
- $Ot_{24} = < t_{24}, p_{27}, t_{21} >$ y $Ot_{25} = < t_{25}, p_{31}, p_{32}, t_{26} >$,
- $CNS(t^8_{pf}) = \{t_{24}, p_{27}, t_{21}, t_{25}, p_{31}, p_{32}, t_{26}\}$.

A continuación se identifican los lugares medibles comunes entre las faltas modeladas: $CNS(t_{of1}) \cap CNS(t_{of2}) = \{p_5\}$, $CNS(t_{of1}) \cap CNS(t^1_{pf}) = \{p_{12}\}$, $CNS(t^1_{pf}) \cap CNS(t^3_{pf}) = \{p_{14}\}$, $CNS(t^2_{pf}) \cap CNS(t^4_{pf}) = \{p_{17}\}$, $CNS(t^3_{pf}) \cap CNS(t^5_{pf}) = \{p_{20}\}$, $CNS(t^4_{pf}) \cap CNS(t^6_{pf}) = \{p_{22}\}$, $CNS(t^5_{pf}) \cap CNS(t^7_{pf}) = \{p_{25}\}$ y $CNS(t^6_{pf}) \cap CNS(t^8_{pf}) = \{p_{27}\}$. El grafo *GCDF* muestra las relaciones encontradas entre las faltas del sistema.

Una vez que el grafo *GCDF* se ha creado se obtiene un conjunto de r particiones de faltas que se utilizan para construir los modelos del diagnosticador distribuido. La solución consiste en agrupar recursivamente todas las faltas conectadas a los vértices con mayor valencia, al considerar todas las faltas dentro de un módulo se evita la duplicación de lugares y por lo tanto la comunicación entre diagnosticadores, posteriormente, la partición encontrada se usa para obtener el número m de modelos diagnosticadores posibles que el usuario requiere.

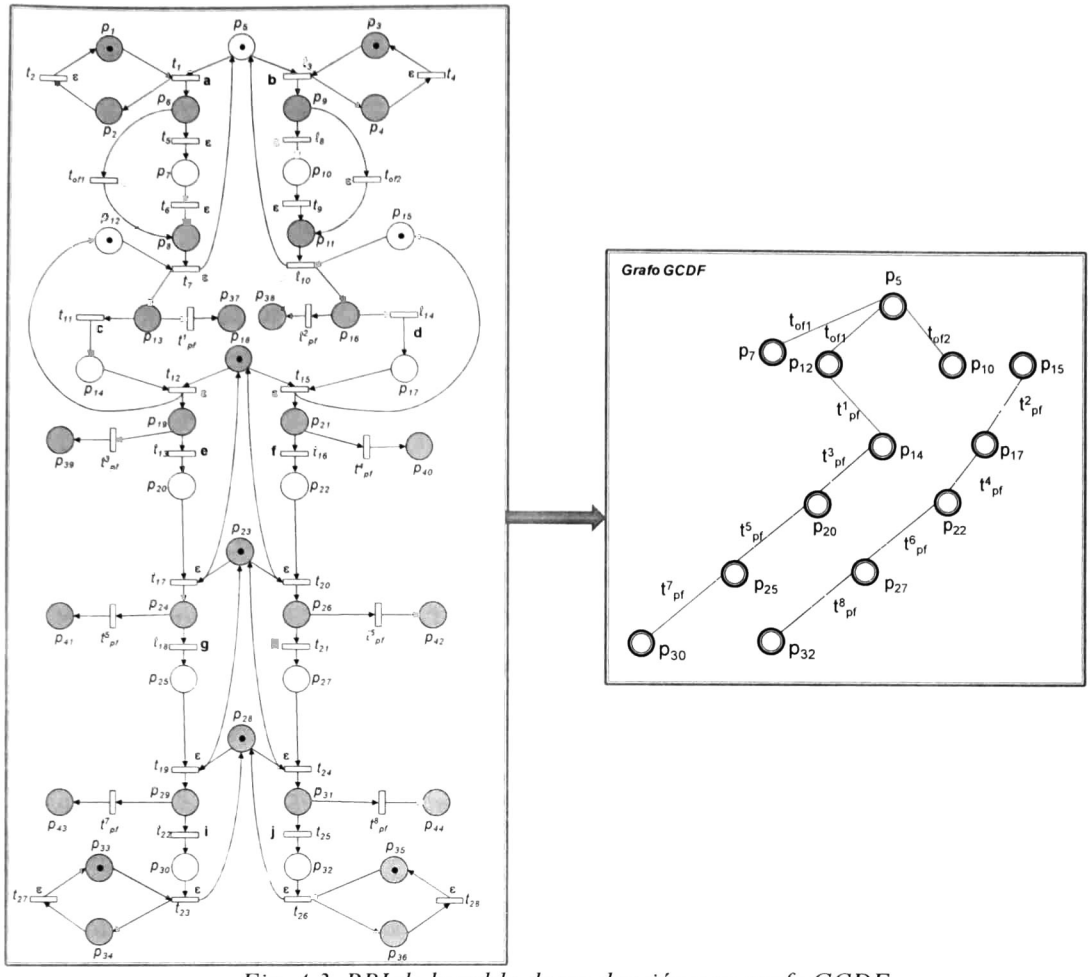

Fig. 4.3. RPI de la celda de producción y su grafo GCDF

4.3. Técnica implementada para modelos diagnosticadores

El algoritmo 4.1 propuesto utiliza el modelo del sistema (Q, M_0) y al grafo *GCDF* para obtener una partición r de faltas.

La partición genera un conjunto de módulos, se usa el grafo *GCDF* para obtener particiones de faltas que cumplen con la proposición 4.8, de ser una partición con comunicación mínima, y da la pauta para obtener la lista de distribuciones con m módulos, donde m va de 2 hasta $|E|$, donde $|E| = |T^{PF}| + |T^{OF}|$.

El algoritmo 4.1, genera una distribución con r módulos para diagnosticadores, y existe comunicación mínima entre los r diagnosticadores.

- Recibe como parámetro el sistema modelado por la *RPI* (Q,M_0) y $GCDF = (V,E)$ un grafo de comunicación dependiente entre faltas de (Q,M_0).
- Se obtienen los vértices de mayor valencia, los cuales significan faltas que tienen dependencia de salidas asociadas a los lugares.
- Se selecciona el vértice de mayor valencia que tenga los vértices adyacentes con menor valencia, es decir, aquella falta mayor que tenga dependencia con la falta de menor.

- Una vez que se ha elegido al vértice, se construyen los módulos de la distribución tomando en cuenta las aristas conectadas al vértice, las aristas representan las faltas y el conjunto de nodos soporte de las faltas tienen los lugares y transiciones del sistema, por lo tanto se obtiene como salida un módulo de la distribución, y una partición de los nodos del grafo *GCDF*, donde los nodos elegidos es la partición *C* y el conjunto de vértices restantes son los descartados.

Algoritmo 4.1. Obtención de una distribución con *r* módulos diagnosticadores

Entrada: (Q,M_0) y $GCDF = (V,E)$

Salida: $DN_x = \{\mu_1,\ldots,\mu_m\}$, C – Representa los vértices elegidos para formar los módulos de DN_x

1. $C = \varnothing$
2. $DN_x = \varnothing$
3. NOD = $\{x_i \mid x_i \in V\}$
4. Mientras E $\neq \varnothing$ realiza lo siguiente
   - 4.1. $SC \subseteq$ NOD, tal que z $\in SC$ $d(z) = \max_x d(g)$, $g \in NOD$
     - $\forall v_i \in SC$, se define $SC_i' = \{w \mid w$ es un vértice adyacente a $v_i\}$
   - 4.2. Obtener el conjunto $R = \{v_k \mid v_k \in SC$, tal que $\exists v_j \in SC_i'$ y $d(v_j) = \min_b d(b), b \in \bigcup_i SC_i'\}$
     - $\forall v_k \in R$, $\mu_x = \cup\, CNS(F)$, $F = \{f_j \mid f_j \in E$ son aristas incidentes a $v_k\}$
     - $DN_x = DN_x \cup \mu_x$
     - $E = E - F$
     - NOD = NOD $- (R \cup v_j \in SC_i'$ y $d(v_j) = \min_b d(b), b \in \bigcup_i SC_i')$
     - $C = C \cup R$
5. Fin Mientras

El algoritmo anterior consta de dos pasos importantes: en el primero se obtienen el conjunto de vértices que son claves para formar los módulos, así como los vértices adyacentes. El grado de los vértices representa el total de faltas que existe con mayor dependencia y agruparlas significa comunicación mínima. Si se quiere obtener una cantidad menor o mayor de particiones su comunicación aumentará o disminuirá según sea el caso. El segundo paso forma los módulos utilizando el conjunto soporte de las faltas asociadas a las aristas del vértice elegido y establece la comunicación entre cada diagnosticador.

Ejemplo 4.4. En el ejemplo 4.2, se presentó el modelo del sistema con el correspondiente grafo de comunicación de las faltas. Los dos modelos son el parámetro de entrada para el algoritmo 4.1, para encontrar la distribución con menor comunicación entre módulos. El algoritmo se comportaría de la siguiente manera:

$C = \varnothing$,

$DN_x = \varnothing$,

NOD = $\{p_1, p_3, p_5\}$, entra al ciclo y como el total de aristas son 2 (indicando el total de faltas del sistema), realiza lo siguiente:

$SC = \{p_1\}$ ya que es el vértice con mayor valencia, $\delta(p_1) = 2$,

$SC_1 = \{p_3, p_5\}$,

$R = \{p_1\}$ y $\mu_1 = (Q^d, M^d{}_0)$, $C = \{p_1\}$ y termina la ejecución del algoritmo, debido a que el modelo del sistema es pequeño, nos dice que las dos faltas tienen que estar relacionadas para que la comunicación que exista sea la mínima, es decir, que haya cero comunicación, el modelo es muy pequeño y por lo tanto no sería óptimo distribuirlo, pero si se llegará a realizar una partición, pues se incrementa la comunicación en 1 mensaje, ya que el lugar p_1 está compartido por las faltas del

sistema. En la figura 4.7, se presenta el resultado de aplicar el algoritmo 4.1 para obtener una distribución óptima.

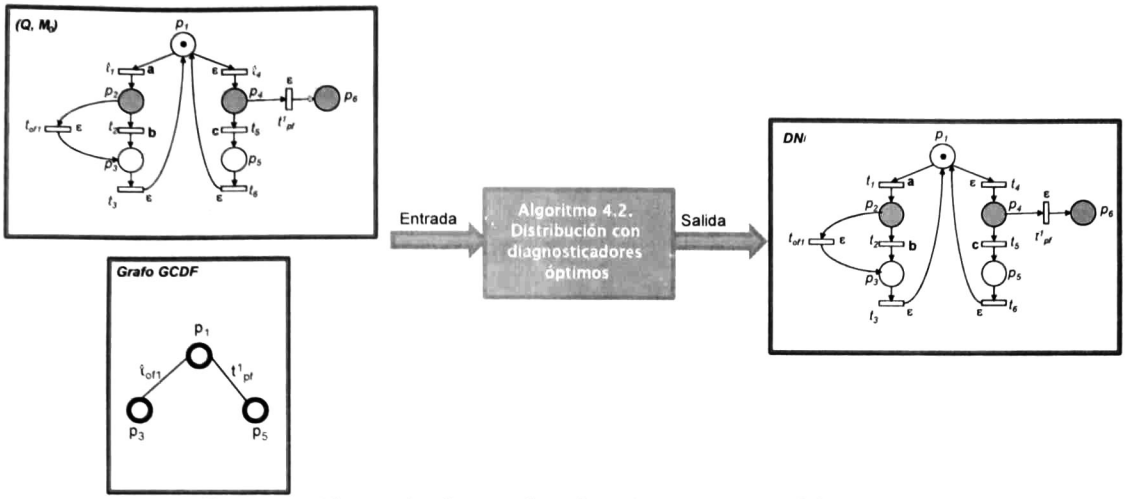

Fig. 4.4. Obtención de una distribución para un modelo pequeño

Ejemplo 4.5. A continuación se presenta la ejecución del algoritmo de la distribución para la celda de producción que se muestra en la figura 2.7. El algoritmo se comportaría de la siguiente manera:
$R = \varnothing$,
$DN_x = \varnothing$,
NOD = {p_5, p_7, p_{10}, p_{12}, p_{14}, p_{15}, p_{17}, p_{20}, p_{22}, p_{25}, p_{27}, p_{30}, p_{32}}, entra al ciclo y como el total de aristas son 10 (considerando a las aristas repetidas como una),
$E = \{t_{of1}, t_{of2}, t^1_{pf}, t^2_{pf}, t^3_{pf}, t^4_{pf}, t^5_{pf}, t^7_{pf}, t^6_{pf}, t^8_{pf}\}$, realiza lo siguiente:
SC = {p_5, p_{12}, p_{14}, p_{17}, p_{20}, p_{22}, p_{25}, p_{27}} ya que son los vértices con mayor valencia, $\delta(p_i) = 2$, $p_i \in SC$. Se toma el vértice p_5 ya que al obtener el conjunto de vértices adyacentes:
SC_5 = {p_7, p_{10}, p_{12}, p_{15}} es el que tiene mayor cantidad de vértices con menor valencia, esto es, $\delta(p_7) = \delta(p_{10}) = 1$.
R = {p_5}
C = {p_5}
*Un módulo tiene el conjunto de nodos soporte asociado a las faltas (t_{of1}, t_{of2}), es decir,
$\mu_1 = CNS(t_{of1}) \cup CNS(t_{of2})$
$E = \{t^1_{pf}, t^2_{pf}, t^3_{pf}, t^4_{pf}, t^5_{pf}, t^7_{pf}, t^6_{pf}, t^8_{pf}\}$,
NOD = {p_{12}, p_{14}, p_{15}, p_{17}, p_{20}, p_{22}, p_{25}, p_{27}, p_{30}, p_{32}}.
SC = {p_{14}, p_{17}, p_{20}, p_{22}, p_{25}, p_{27}},
SC_{14}= {p_{12}, p_{20}},
SC_{17}= {p_{15}, p_{22}},
SC_{20}= {p_{14}, p_{25}},
SC_{22}= {p_{17}, p_{27}},
SC_{25}= {p_{20}, p_{30}},
SC_{27}= {p_{27}, p_{32}},
R = {p_{14}, p_{17}, p_{25}, p_{27}}, ya que todos estos vértices tienen como adyacentes a vértices de valencia 1.
*Un módulo tiene a las faltas permanentes (1 y 3),
$\mu_2 = CNS(t^1_{pf}) \cup CNS(t^3_{pf})$,
$E = \{t^2_{pf}, t^4_{pf}, t^5_{pf}, t^7_{pf}, t^6_{pf}, t^8_{pf}\}$,
NOD = {p_{15}, p_{17}, p_{20}, p_{22}, p_{25}, p_{27}, p_{30}, p_{32}}.
C = {p_5, p_{14}}

*Un módulo tiene a las faltas permanentes (2 y 4),
$\mu_3 = CNS(t^2_{pf}) \cup CNS(t^4_{pf})$,
$E = \{t^5_{pf}, t^7_{pf}, t^6_{pf}, t^8_{pf}\}$,
NOD = $\{p_{20}, p_{22}, p_{25}, p_{27}, p_{30}, p_{32}\}$.
C = $\{p_5, p_{14}, p_{17}\}$
*Un módulo tiene a las faltas permanentes (5 y 7),
$\mu_4 = CNS(t^5_{pf}) \cup CNS(t^7_{pf})$,
$E = \{t^6_{pf}, t^8_{pf}\}$,
NOD = $\{p_{22}, p_{27}, p_{32}\}$.
C = $\{p_5, p_{14}, p_{17}, p_{25}\}$
*Un módulo tiene a las faltas permanentes (6 y 8),
$\mu_5 = CNS(t^6_{pf}) \cup CNS(t^8_{pf})$,
$E = \varnothing$
NOD = $\varnothing$
C = $\{p_5, p_{14}, p_{17}, p_{25}, p_{27}\}$

El algoritmo indica una partición de 5 módulos diagnosticadores con un total de 3 mensajes (ver figura 4.5). A partir de la partición de vértices del algoritmo es posible crear una lista, para encontrar el conjunto de distribuciones con *m* módulos diagnosticadores según el usuario requiera.

Por la proposición 4.8 se sabe que la cantidad más grande de módulos en una distribución para la celda de producción que se muestra en la figura 2.7 es de 10 y requiere 8 lugares de comunicación para que los módulos sean diagnosticables. En la tabla 4.1 se visualiza una lista que toma este resultado para generar la cantidad mínima de lugares por distribución con *m* módulos, y se verifica que el resultado proporcionado por el algoritmo 4.1, es decir, una distribución con *5* módulos cumple con la cantidad mínima de 3 lugares de comunicación.

Módulos por distribución (m)	Total de faltas (t_{faltas})	Lugares de Comunicación por distribución (P^i_{com})
1	10	0
2	10	0
3	10	1
4	10	2
5	10	3
6	10	4
7	10	5
8	10	6
9	10	7
10	10	8

Tabla 4.1. Lista de particiones posibles y comunicación mínima

Si se desea obtener una distribución diferente a la obtenida por el algoritmo 4.1, la partición *C* generada por el algoritmo 4.1 y el grafo *GCDF* sirven como guía para agregar o quitar módulos. El algoritmo 4.2 utiliza los siguientes criterios: para disminuir la partición arrojada por el algoritmo y obtener una cantidad *m* de módulos menor, se agrupan módulos con vértices de menor valencia que fueron seleccionados, los cuales compartan aristas y para aumentar la cantidad *m* se particionan vértices que tengan una cantidad de faltas mayor, con lo cual se sigue respetando comunicación mínima.

El algoritmo 4.2, muestra cómo obtener una distribución con m módulos diagnosticadores, donde $m \neq r$, el valor máximo que puede tener m es la cantidad total de aristas del grafo *GCDF*.

- Recibe como entrada el grafo *GCDF*, la distribución generada DN_x con r módulos y la partición de vértices $C \subseteq V$ generados por el algoritmo 4.1.

- Se obtiene primero una diferencia de módulos, si $r > m$ significa una diferencia positiva, la cual indica que los módulos obtenidos por el algoritmo 4.1. son mayores que los que el usuario desea encontrar, por lo tanto se obtendrá una nueva distribución, agrupando los vértices que se consideran en C Si $r < m$ significa una diferencia negativa, indicando que existe una cantidad mayor de módulos que se desean, por lo tanto se obtendrá una nueva distribución considerando una división de los módulos generados por el algoritmo 4.1, considerando aquellos vértices con mayor valencia, de tal forma que existirá más vértices del grafo en el conjunto C.

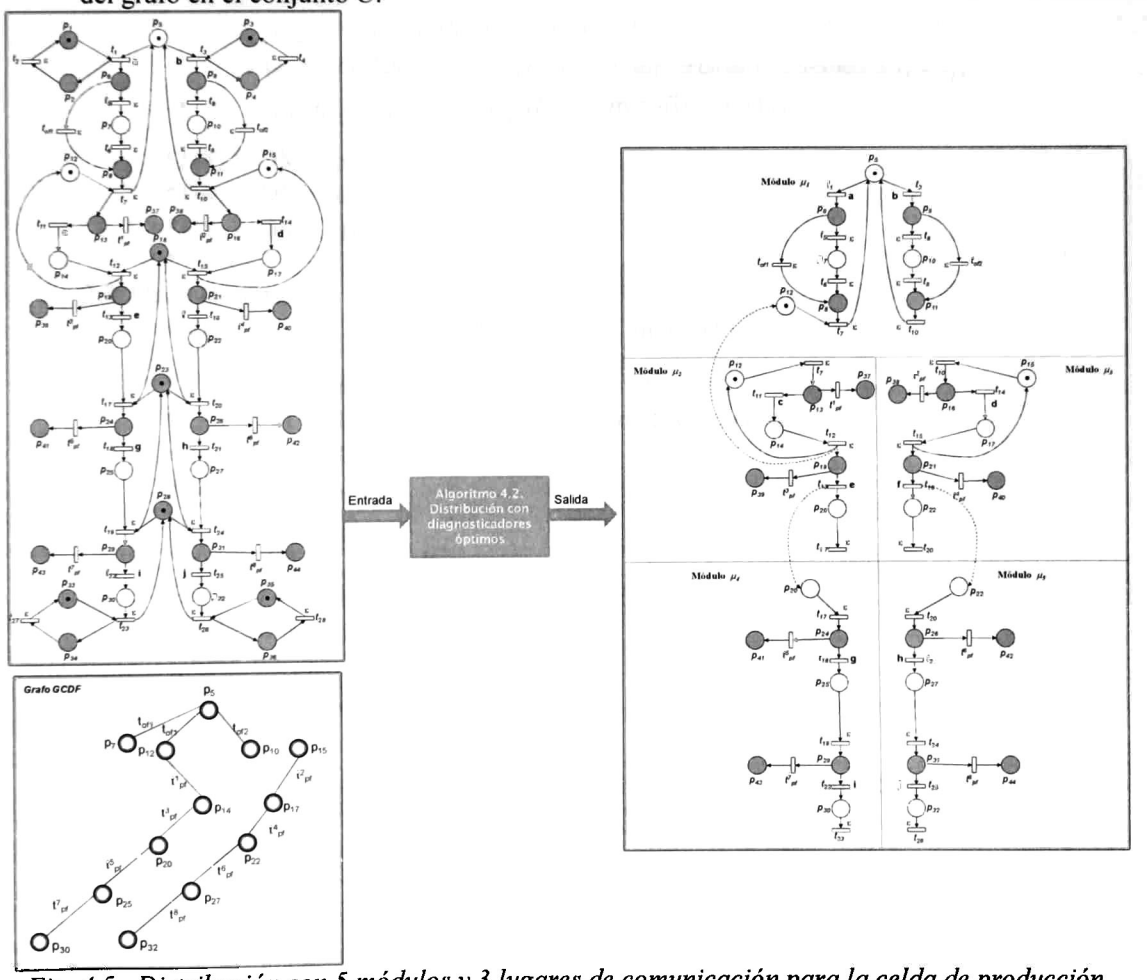

Fig. 4.5. Distribución con 5 módulos y 3 lugares de comunicación para la celda de producción

Algoritmo 4.2. Obtención de distribución con m módulos

Entrada: $RCS = (V,E)$, $DN_x = \{\mu_1,\ldots,\mu_r\}$, C – Representa los vértices elegidos para formar los módulos de DN_x

Salida: $DN_x = \{\mu_1,\ldots,\mu_m\}$,

1. Si $m \leq |E|$ entonces
2. Mientas $r \neq m$ realiza lo siguiente
   2.1. dif = $r - m$
   2.2. Si dif > 0 entonces
      * $\forall v_j \in C,\ d(v_j) = \min_b d(b),\ b \in \bigcup_i V$
      * $\mu_{ant1} = \cup\ CNS(F),\ F = \{f_j \mid f_j \in E$ son aristas incidentes a $v_j\}$
      * Si $\exists\ v_i \in C$, tal que, v_i comparte una arista con un vértice adyacente a v_j entonces
         - $\mu_{ant2} = \cup\ CNS(F),\ F = \{f_j \mid f_j \in E$ son aristas incidentes a $v_i\}$
         $\mu_x = \cup\ CNS(F),\ F = \{f_j \mid f_j \in E$ son aristas incidentes a v_i y $v_j\}$
      * De lo contrario
         - $\mu_{ant2} = \cup\ CNS(F),\ F = \{f_j \mid f_j \in E$ son aristas incidentes a $v_i\}$
         $\mu_x = \cup\ CNS(F),\ F = \{f_j \mid f_j \in E$ son aristas incidentes a v_i y $v_j\}$
         - donde $v_i \neq v_j$ y $v_i \in C,\ d(v_i) = \min_b d(b),\ b \in \bigcup_i C_i$
      * $DN_x = DN_x - \mu_{ant1}$
      * $DN_x = DN_x - \mu_{ant2}$
      * $DN_x = DN_x \cup \mu_x$
      * $r = r - 1$
   2.3. De lo contrario
      * $\forall v_j \in C,\ d(v_j) = \max_b d(b),\ b \in \bigcup_j C_j$ y $\max_b d(b) > 1$
      * $F = \{f_j \mid f_j \in E$ son aristas incidentes a $v_j\}$
      * $\mu_{ant} = \cup\ CNS(F),\ F = \{f_j \mid f_j \in E$ son aristas incidentes a $v_j\}$
      * $F_i \subseteq F \mid |F_i| \leq |F|/2$
      * $F_j \subseteq F \mid |F_j| \leq |F|/2$
      * $\mu_x = \cup\ CNS(F_i)$
      * $\mu_y = \cup\ CNS(F_j)$
      * $DN_x = DN_x - \mu_{ant}$,
      * $DN_x = DN_x \cup \mu_x \cup \mu_y$
      * $r = r + 1$
   2.4. Fin del si
3. Fin Mientras
4. De lo contrario
   4.1. Escribe mensaje "El total de módulos es cuando mucho la cantidad de faltas"
   4.2. $m = |E|$
   4.3. Regresa al paso 2
5. Fin del si

Ejemplo 4.6. Se desea encontrar *3* módulos para una implementación con 3 diagnosticadores para la celda de producción que se muestra en la figura 2.7. En la figura 4.5 se muestra una distribución con *5* módulos y existe una partición de vértices C = {p_5, p_{14}, p_{17}, p_{25}, p_{27}}, las cuales son obtenidas mediante el algoritmo 4.1. Se obtiene una diferencia positiva de módulos, lo cual significa unir módulos a partir los vértices obtenidos en C, debido $\forall v_i \in C$, $d(v_i) = 2$, entonces los vértices tienen la misma probabilidad de ser elegidos, por lo tanto, se inicia con el primero que aparece en la lista. Generándose, dos nuevos módulos formados por la unión del conjuntos soportes de las aristas

incidentes a los vértices: p_5 y p_{14} para un nuevo módulo y el otro módulo generado por: p_{17} y p_{27}, y un tercer módulo generado por las aristas conectadas a p_{25}, por lo tanto, es posible tener un total de 3 módulos con 1 lugar de comunicación, como lo señala la distribución presentada en la figura 4.6, y además, la distribución cumple con el criterio de la tabla 4.1, por lo tanto, es una distribución con tres módulos y cantidad mínima de lugares de comunicación.

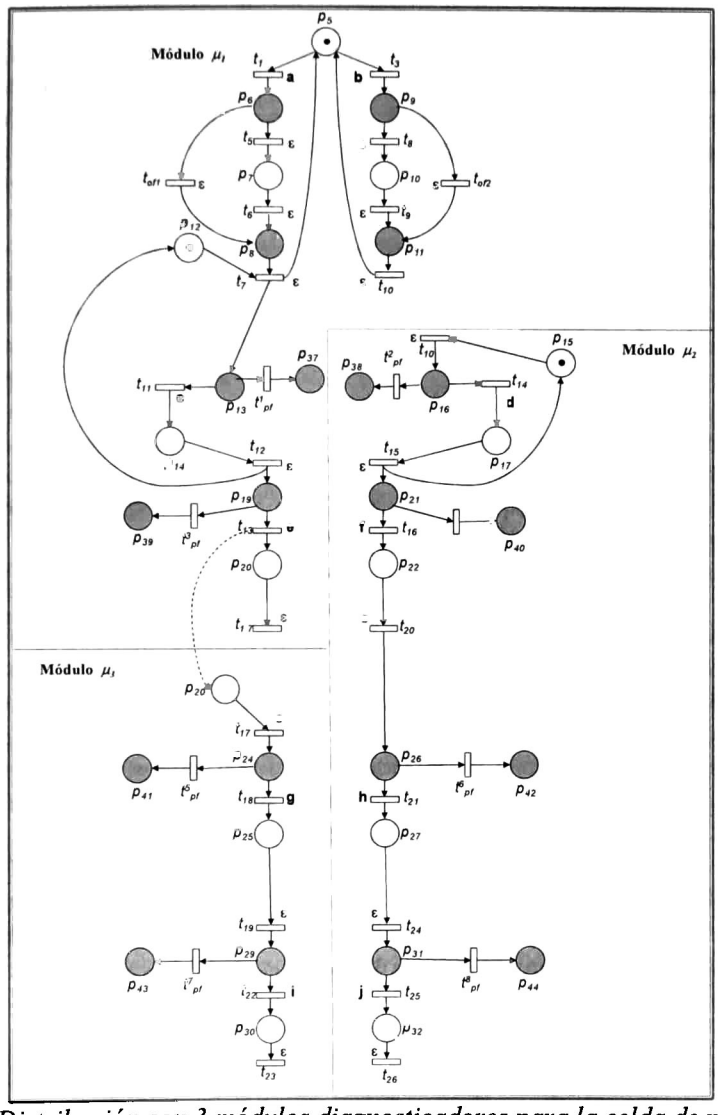

Fig. 4.6. Distribución con 3 módulos diagnosticadores para la celda de producción

Conclusiones

En esta tesis se abordó el problema del diagnostico distribuido de sistemas de eventos discretos, enfocándose en los aspectos de la diagnosticabilidad distribuida, la confiabilidad de la red de diagnosticadores y en la operación eficiente del conjunto de diagnosticadores.

El estudio de la diagnosticabilidad distribuida dio lugar a la extensión de una caracterización previa de diagnosticabilidad estructural sobre el modelo global, estableciendo las condiciones suficientes para sistemas de eventos discretos parcialmente observables. Esto permitió diagnosticar una clase mayor de modelos. Posteriormente la caracterización de un modelo distribuido se expresó en términos de la nueva caracterización.

El esquema propuesto de diagnosticadores distribuidos con modelos redundantes permite la detección de fallas en los diagnosticadores. Los esquemas con doble redundancia y triple redundancia modular introducidos en la arquitectura aumentan la confiabilidad en la operación de la red de diagnosticadores, permitiendo mantener la función de monitoreo cuando algún diagnosticador quede fuera de operación.

El método propuesto para la descomposición del modelo global permite la obtención de sub-modelos de manera eficiente, logrando reducir al mínimo el número de mensajes entre los diagnosticadores. Soportado por el grafo de comunicación dependiente entre faltas y un adecuado procesamiento de éste, se ha podido resolver el problema combinatorio inherente a la descomposición de grafos.

El trabajo realizado contiene avances respecto a los trabajos propuestos en la literatura. Sin embargo durante la construcción de soluciones para los problemas abordados, se detectan algunos aspectos del trabajo que merecen ser estudiados de manera más extensa y profunda, a saber:

- Relajar las condiciones de la caracterización estructural del modelo global diagnosticable del cual parte la arquitectura distribuida para extender la clase de *RPI* diagnosticables.
- Proponer un modelo adicional que especifique el comportamiento seguro del sistema, para evaluar si las fallas en los componentes del sistema son detectadas dentro de un comportamiento seguro.
- Definir la propiedad de diagnosticabilidad distribuida sin necesidad de la construcción del modelo global del sistema ni de su análisis de diagnosticabilidad.
- Obtener el conjunto de diagnosticadores distribuidos óptimos con sub-modelos redundantes.

Referencias

[Alcaraz-Mejía, et al., 2003] M. Alcaraz-Mejía, E. López-Mellado, Antonio Ramírez-Treviño, I. Rivera Rangel. "Petri Net Based Fault Diagnosis of Discrete Event Systems", Proc. of the IEEE International Conference on Systems, Man and Cybernetics. pp. 4730-4735. 2003.

[Arámburo-Lizárraga, et al., 2005] J. Arámburo-Lizárraga, E. López-Mellado, A. Ramírez-Treviño. "Distributed Fault Diagnosis using Petri Net Reduced Models". Proc. of the IEEE International Conference on Systems, Man and Cybernetics. pp. 702-707. 2005.

[Arámburo-Lizárraga, et al., 2007a] J. Arámburo-Lizárraga, E. López-Mellado, A. Ramírez-Treviño. "Design of Low Interaction Distributed Diagnosers for Discrete Event Systems" Proc. of the 4th Int. Conf. on Informatics in Control, Automation and Robotics. pp. 189-194. 2007.

[Aramburo-Lizárraga, et al., 2007b] J. Arámburo-Lizárraga, E. López-Mellado, A. Ramírez-Treviño, E. Ruiz-Beltrán. "Reliable Distributed Fault Diagnosis using Redundant Diagnosers" Proc. of the 1st IFAC Workshop on Dependable Control of Discrete Systems. Cachan, Francia. 2007.

[Aramburo-Lizárraga, et al., 2008a] J. Arámburo-Lizárraga, A. Ramírez-Treviño, E. López-Mellado, E. Ruiz-Beltrán. "Fault Diagnosis in Discrete Event Systems using Interpreted Petri Nets" Capítulo 5° del libro Advances in Robotics, Automation and Control. Publicado por In-Teh. pp. 69-84. 2008.

[Aramburo-Lizárraga, et al., 2008b] J. Arámburo-Lizárraga, A. Ramírez-Treviño, E. López-Mellado. "Fault Diagnosis in Discrete Event Systems using Distributed Diagnosers". IV Semana Nacional de Ingeniería Electrónica. SENIE08. pp. 550-557. 2008.

[Bourdeaud'huy y Toguyeni, 2006] T. Bourdeaud'huy y A. Toguyeni. "A Petri-Net based approach for the reconfiguration of flexible manufacturing systems using optimization techniques" INCOM'06. 2006.

[Cassandras y Lafortune, 2008] C.G. Cassandras y S. Lafortune. "Introduction to Discrete Event Systems". Springer, 2008.

[Chen y Patton, 1999] J. Chen y R.J Patton. "Robust Model-Based Fault Diagnosis for Dynamic Systems" Asian Studies in Computer Science and Information Science. Kluwer Academic Publishers, Boston. 1999.

[Chung, et al., 2003] S.L Chung, C. Chung-Wu, and M. Jeng. "Failure Diagnosis: A case study on Modeling and Analysis by Petri Nets". Proceedings of the IEEE Conference on Systems, Man & Cybernetics. pp. 2727-2732, 2003.

[Da Silveria, et al., 2002] M. Da Silveira, M. Combacau y A. Subias. "From centralized to distributed models: A systematic procedure based on Petri nets" SMC'02. IEEE International Conference on Systems, Man and Cybernetics, Vol. 1. 2002.

[Debouk, et al., 2000] R. Debouk, S. Lafortune y D. Teneketzis. "Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems", Kluwer Academic Publishers, Discrete Event Systems: Theory and Applications, vol. 10. pp. 33-79. 2000.

[Fabre, et al., 2000] E. Fabre, A. Benveniste, C. Jard, L. Ricker y M. Smith. "Distributed

State Reconstruction for Discrete Event Systems" In 39th IEEE Conference on Decision and Control (CDC), Sydney. pp. 2252-2257. 2000.

[Fisher, et al., 1985] M. J. Fischer, N. A. Lynch y M. S. Paterson. "Impossibility of Distributed Consensus with One Faulty Process". Journal of the ACM, 32(2). pp.374-382. 1985.

[Genc y Lafortune, 2003] S. Genc y S. Lafortune. "Distributed Diagnosis of Discrete-Event Systems Using Petri Nets", 24th International Conference on Application and Theory of Petri Nets, ICATPN 2003, Eindhoven, The Netherlands. Lecture Notes in Computer Science, vol. 2679. pp. 316-336. 2003.

[Genc y Lafortune, 2005] S. Genc y S. Lafortune. "A Distributed Algorithm for On-Line Diagnosis of Place-Bordered Nets", presentado en la session titulada Dependable Manufacturing Systems I, 16th IFAC World Congress, Praha, Czech Republic. 2005.

[Genc y Lafortune, 2007] S. Genc y S. Lafortune. "Distributed Diagnosis of Place-Bordered Petri Nets" IEEE Transactions on Automation Science and Engineering. 2007.

[Giua y Seatzu, 2005] A. Giua and C. Seatzu. "Fault detection for discrete event systems using Petri nets with unobservable transitions". In the 44th International Conference on Decision and Control and European Control Conference, pp. 6323-6328. 2005.

[Haar, et al., 2005] S. Haar, A. Benveniste, E. Fabre y C. Jard. "Fault Diagnosis for Distributed Asynchronous Dynamically Reconfigured Discrete Event Systems" 16th IFAC World Congress, Praha, Czech Republic. 2005.

[Hashtrudi, et al., 2003] S. Hashtrudi Zad, R. H. Kwong y W.M. Wonham. "Fault Diagnosis in Discrete-Event Systems: Framework and Model Reduction" IEEE Transactions on Automatic Control, vol. 48, no. 7, pp.1199-1211, 2003.

[Jalote, 1994] P. Jalote. Fault Tolerance in distributed systems. Prentice Hall. 1994.

[Jiang, et al , 2001] S. Jiang, Z. Huang, V. Chandra and R. Kumar. "A polynomial algorithm for testing diagnosability of Discrete-Event Systems" IEEE Transactions on Automatic Control, vol. 46, no. 8. pp. 1318-1321, 2001.

[Jiang, et al , 2003] S. Jiang, R. Kumar y H.E.García. "Optimal Sensor Selection for Discrete-Event Systems with Partial Observation". IEEE Trans. on Automation, Control, vol., 48, no. 3. pp. 369-381. 2003.

[Jiroveanu y Boel, 2005] G. Jiroveanu y R. K. Boel. "Petri Net model-based distributed diagnosis for large interacting systems". Proceedings of the 16th International Workshop on Principles of Diagnosis (DX'05). pp. 25-30. 2005.

[Koh y DiCesare, 1991] J. Koh y F. DiCesare. "Modular transformation methods for generalized Petri Nets and their application to automated manufacturing systems" IEEE Transactions on Systems, Man & Cybernetics, pp. 1512-1522. 1991.

[Lampson, 1993] B.W. Lampson. "Reliable Messages". In S. Mullender, (ed), Distributed Systems: Architecture and Implementation, chapter 10. Addison-Wesley. 1993.

[Nissanke, 1997] N. Nissanke. "Realtime Systems" Prentice-Hall. Primera Edición. 1997.

[Pan y Hashtrudi-Zad, 2007] J. Pan y S. Hashtrudi-Zad. "Diagnosability Analysis and Sensor Selection in Discrete-Event Systems with Permanent Failures". Proceedings of the 3rd Annual IEEE Conference on Automation Science

and Engineering. Scottsdale, AZ, USA. 2007.

[Pradhan, 1996] D.K. Pradhan. "Fault-Tolerant Computer System Design". Prentice Hall. 1996.

[Provan, 2002] G. Provan. "Distributed Diagnosability Properties of Discrete Event Systems" Proc. Of the American Control Conference. Anchorage, AK. 2002.

[Qui y Kumar, 2005] W. Qiu. y R. Kumar. "Decentralized Diagnosis of Event-Driven Systems for Safely Reacting to Failures" IFAC World Congress, Praga. 2005.

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

[Ramirez-Treviño, et al., 2004] A. Ramírez-Treviño, E. Ruiz Beltrán, I. Rivera-Rangel, E. López-Mellado. "Diagnosability of Discrete Event Systems. A Petri Net Based Approach" Proc. of the IEEE International Conference on Robotic and Automation. pp. 541-546. 2004.

[Ramirez-Treviño, et al., 2007] A. Ramírez-Treviño, E. Ruiz-Beltrán, I. Rivera-Rangel y E. López-Mellado. "Online Fault Diagnosis of Discrete Event Systems. A Petri Net Based Approach". IEEE Transactions on Automation Science and Engineering, vol. 4, no. 1. pp. 31-39. 2007.

[Rivera-Rangel, 2004] I. Rivera-Rangel. "Observability and modular synthesis of Petri net models of Discrete Event Systems" Tesis Doctoral. Centro de Investigación y Estudios Avanzados del IPN Unidad Guadalajara. Guadalajara, Jalisco. 2004.

[Ru y Hadjicostis, 2007] Y. Ru y Ch. N. Hadjicostis. "Approximating Optimal Place Sensor Selection for Structural Observability in Discrete Event Systems Modeled by Petri Nets" Proceedings of the 46th IEEE Conference on Decision and Control. 2007.

[Ruiz-Beltrán, 2007] E. Ruiz-Beltrán. "Esquemas de Diagnóstico de Sistemas de Eventos Discretos" Tesis Doctoral. Centro de Investigación y Estudios Avanzados del IPN Unidad Guadalajara. Guadalajara, Jalisco. 2007.

[Ruiz-Beltrán, et al., 2004] E. Ruiz-Beltrán, A. Ramírez-Treviño y E. López-Mellado. "Building Diagnosable Petri Net Models for Distributed Fault Location of Discrete Event Systems" Proceedings of the International Conference on Systems, Man, & Cybernetics. pp. 4929-4934. 2004.

[Ruiz-Beltrán, et al., 2005] E. Ruiz-Beltrán, I. Jiménez, A. Ramírez, E. López, M. Meda. "Fault detection and location in DES using Petri Nets". Proceedings of the 3rd. Annual IEEE Conference on Automation and Engineering. pp. 1645-1650. 2005.

[Ruiz-Beltrán, et al., 2006] E. Ruiz-Beltrán, E. López-Mellado y A. Ramírez. "Fault Diagnosis based on Petri net reduced models" Proceedings of the 3rd. International Conference on Electrical and Electronics Engineering and XII Conference on Electrical Engineering. pp. 222-226. 2006.

[Ruiz-Beltrán, et al., 2007] E. Ruiz-Beltrán, A. Ramírez y E. López. "A Structural Characterization of Diagnosable Petri Net Models". Proceedings of the IEEE Conference on Systems, Man and Cybernetics. pp. 1137-1142. 2007.

[Sampath, et al., 1995] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen y D. Teneketzis. "Diagnosability of discrete event systems" IEEE Transactions on Automatic and Control, vol 4, no. 9. pp. 1555-1575. 1995.

[Sampath, et al., 1996] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen y D. Teneketzis. "Failure Diagnosis Using Discrete-Event Models" IEEE Transactions on Control System Technology, vol. 4, no. 2. pp. 105-207. 1996.

[Siewiorek y Swarz, 1992] D.P. Siewiorek y R.S. Swarz. "Reliable computer systems: Design and Evaluation" Digital Press. 1992.

[Silva, 1985] M. Silva. "Las redes de Petri: en la automática y la informática" AC, Madrid, España. 1985.

[Su, et al., 2002] R. Su, W.M. Wonham, J. Kurien, X. Koutsoukos. "Distributed Diagnosis for Qualitative Systems". In Proc. 6th International Workshop on Discrete Event Systems (WODES'02), Zaragoza, España. pp. 169-174. 2002.

[Su y Wonham, 2004] R. Su y W.M. Wonham. "A Model of Component Consistency in Distributed Diagnosis" In Proc. 7th IFAC International Workshop on Discrete Event Systems, Reims, Francia. pp. 427-432. 2004.

[Yoo y Lafortune, 2002] T.S. Yoo y S. Lafortune. "Polynomial-time verification of diagnosability of partially-observed discrete-event systems," IEEE Trans. Automatic Control 47(9). pp.1491-1495. 2002.

[Yoo y Lafortune, 2002] T.S. Yoo y S. Lafortune. "NP-completeness of sensor selection problems arising in partially observed discrete event systems". IEEE Trans. on Automation, Control, vol., 47, no. 9. pp. 1495-1499. 2002.

[Zhou y DiCesare, 1993] M. C. Zhou y F. DiCesare. "Petri Net Synthesis for Discrete Event Control of Manufacturing Systems". Kluwer Academic Publishers, Boston, MA. 1993.

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Diagnóstico Distribuido Confiable de Sistemas de Eventos Discretos

del (la) C.

Jesús ARÁMBURO LIZÁRRAGA

el día 12 de Junio de 2009.

Dr. Juan Manuel Ramírez Arredondo
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Antonio Ramírez Treviño
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González Pico
Investigador CINVESTAV 2A
CINVESTAV Unidad Guadalajara

Dra. María Elena Meda Campaña
Profesor Titular
Universidad de Guadalajara CUCEA