

AUTOMATIZACIÓN DEL CONOCIMIENTO PARA DIAGNÓSTICO AUTOMÁTICO DE FALLAS EN CENTRALES DE GENERACIÓN HIDRAÚLICA MEDIANTE REDES DE PETRI

Germán Zapata Madrigal
Universidad Nacional de Colombia Sede Medellín
Facultad de Minas
Medellín, Colombia
gdzapata@unal.edu.co

Evelin Tobón Montaña
Universidad Nacional de Colombia Sede Medellín
Facultad de Minas
Medellín, Colombia
ejtobon@unal.edu.co

José Iván Velásquez Duque
EMGESA S.A E.S.P
División Técnica
Bogotá, Colombia
jvelazqu@emgesa.com.co

Luis Eder Marín
EMGESA S.A E.S.P
División Técnica
Bogotá, Colombia
lmarin@emgesa.com.co

Resumen— En el diagnóstico de fallas para una central de generación hidráulica, se presentan decisiones de los operadores que pueden ser modeladas mediante sistemas de eventos discretos. Si se tiene un modelo estructurado del conocimiento del operador, se facilita la automatización del proceso de diagnóstico y en consecuencia, se pueden reducir los errores humanos y mejorar los tiempos de respuesta para analizar y localizar las fallas. Se propone una herramienta informática para el diagnóstico de fallas y automatización del conocimiento, basada en redes de Petri que utiliza como núcleo un motor que ejecuta la red a partir de la ecuación de estado, y un editor que permite agregar nuevos modelos para aumentar la base de conocimiento. La herramienta y los modelos capturan las interacciones que tiene el operador con los sistemas de protección y control de la central. El sistema fue implementado para una central de generación hidráulica de EMGESA.

Keywords- *Diagnóstico de fallas en generación de energía, automatización del conocimiento, redes de petri, modos de falla.*

I. INTRODUCCIÓN

El diagnóstico de fallas es una funcionalidad importante y crítica que deben ejecutar los sistemas de supervisión de los procesos industriales. Las fallas generan estados de no disponibilidad de la infraestructura de producción que pueden generar retrasos e incumplimientos en los compromisos adquiridos.

El diagnóstico de las fallas es generalmente una tarea compleja que requiere amplio conocimiento y experiencia por parte del personal que supervisa el proceso. Los desarrollos que se realicen para apoyar estas tareas contribuyen a disminuir los tiempos de parada y los errores humanos que se cometen en las operaciones de restablecimiento.

Muchas de las decisiones y del conocimiento que tienen los operadores tienen características de eventos discretos y pueden ser modeladas con herramientas como las redes de Petri. Si se tienen modelos estructurados del conocimiento, las decisiones modeladas se pueden automatizar.

Una aplicación del concepto de automatización del conocimiento para diagnóstico de fallas en centrales de generación hidráulica, haciendo uso de las redes de Petri para la realización de los modelos de conocimiento, se ha desarrollado conjuntamente por Emgesa S.A E.S.P y la Universidad Nacional de Colombia con la financiación de Colciencias. En este trabajo se describe el aplicativo y la fundamentación teórica del concepto.

II. REVISIÓN BIBLIOGRÁFICA

Los métodos o técnicas del diagnóstico de fallas, hacen uso del conocimiento que se tiene del comportamiento del proceso para detectar sucesos anormales en su funcionamiento e identificar las causas [1, 2].

Este conocimiento necesario del proceso, puede provenir de diversas fuentes tales como el personal de la planta, historiales de funcionamiento del proceso o manuales detallados que describen cómo el proceso opera.

Una vez obtenida esta información, se construye un mecanismo de inferencia el cual permite identificar las causas de ésta [3].

El mecanismo de inferencia que se usa para realizar el diagnóstico puede ser por comparación de modelos (Métodos basados en modelos), o por análisis de tendencias (Métodos

basados en historial); además, los métodos de FD se pueden clasificar de acuerdo a la naturaleza de las variables (discretas, continuas, cualitativas, cuantitativas) que describen los comportamientos de los procesos.

[4] define una clasificación detallada de los métodos de diagnóstico de fallas junto con algunos ejemplos de los más comunes dentro de cada categoría.

En [5] y [6] demuestran las ventajas de aplicar redes de Petri (PN) para modelar de manera unificada, los procesos de control y otros esquemas de conocimiento propio de cada empresa.

En el campo de los sistemas de energía, las redes de Petri han sido utilizadas ampliamente para modelar la configuración de las subestaciones [7], obteniendo resultados importantes en la reducción del tiempo de diagnóstico y mejorando la exactitud del mismo.

A partir de los objetos de las redes de Petri, [8] presenta un método para el modelamiento y representación del conocimiento. El método incluye el reconocimiento de varias clases de situación (normal y anormal) presentes en la operación humana de sistemas.

En [9] se hace uso de las FCPN (Fuzzy Colored Petri Nets) para modelar procedimientos operativos y realizar la representación del conocimiento del estado del sistema combinando técnicas de lógica difusa y la teoría de las redes de Petri.

[10] explica la importancia de representar y automatizar el conocimiento operacional, sin embargo destacan la dificultad en la construcción de modelos de conocimiento, pero reconoce el uso de técnicas computacionales para apoyar esta labor.

Los modelos de la experiencia son el punto de interés en [11] usados para construir la estructura base de conocimiento (KBS) a través de la definición de los tipos y niveles del conocimiento y de igual forma del dominio y control del mismo.

[12] evidencia la importancia de construir una base de conocimiento (KB) adecuada, basados en reglas de representación del conocimiento y haciendo uso del enfoque de las PN para construir el KB incluso en términos analíticos.

Los lenguajes computacionales son diseñados específicamente para representar el conocimiento, soportado sobre modelos matemáticos formales, tal como se define en [13], donde se muestra el uso de recursos computacionales y de dos tipos de lenguajes de programación para representar y procesar el conocimiento, a partir de una estructura algorítmica y de datos, dando lugar así a un motor de inferencia.

Desde principios de los 90's, los motores de software genéricos han sido desarrollados a bajo costo en diferentes campos de la ciencia, obteniendo grandes beneficios [14]. Este tipo de software funciona con modelos particulares creados por expertos del dominio específico. En [15] se presentan aspectos importantes a tener en cuenta cuando se requiere unificar modelos de negocio en un prototipo de software, destacando el potencial que tienen estas herramientas cuando ciertos requisitos cambian en el entorno.

Trabajos en los que se desarrollan motores de PN se presentan en [16], [17], [18].

Los motores de ejecución de PN han sido aplicados en la automatización de flujos de trabajo (*Workflow*) [19], [20], [21] y en la automatización de procesos de negocio (BPM – Business Process Management) [22], [23], [24].

III. FUNDAMENTOS TEÓRICOS

Las redes de Petri (PN) son una herramienta matemática que se ha utilizado ampliamente para modelar los denominados “Sistemas de eventos discretos – DES”, que se caracterizan por tener un espacio de estados discreto y finito y por que la evolución entre los estados se presenta por la ocurrencia de eventos aleatorios [25], [26]. Dentro de este tipo de sistemas se pueden mencionar procesos que presentan comportamientos secuenciales, asíncronos, concurrentes, de recursos compartidos, con sincronizaciones. Formalmente se define una PN autónoma y ordinaria como una tupla $N = \langle P, T, F \rangle$, donde P representa un conjunto finito de lugares y T representa un conjunto finito de transiciones. En la representación de un DES mediante PN, los lugares representan los estados y las transiciones los eventos. Las conexiones entre lugares y transiciones se establecen en el conjunto de arcos $F \subseteq (P \times T) \cup (T \times P)$.

Los conjuntos P, T, F describen la estructura estática de la red.

Para establecer los aspectos dinámicos, se definen los conjuntos de entrada salida. El conjunto de lugares de entrada a la transición t está dado por $t = \{p \in P : (p, t) \in F\}$. De manera análoga se pueden definir los conjuntos t', p, p'. Un marcaje M de una estructura N es una proyección $M : P \rightarrow N \cup \{0\}$, que asigna a cada lugar un número de marcas N. Una estructura N con un marcaje inicial M_0 se denomina una red de Petri marcada y se denota (N, M_0) . Una transición $t \in T$ está sensibilizada en un marcaje M si $\forall p \in t : m(p) \geq 1$. Una transición sensibilizada en un marcaje puede disparar y hace evolucionar el marcaje hacia un marcaje M' siguiendo la siguiente regla:

$$M'(p) = \begin{cases} M(p)+1 & \text{si } p \in t' \setminus t \\ M(p)-1 & \text{si } p \in t \setminus t' \\ M(p) & \text{en otro caso} \end{cases}$$

La estructura de la PN se puede expresar mediante la denominada matriz de incidencia, la cual está dada por $C = B - A$, donde:

$$b_{ij} = \begin{cases} 1 & \text{si } (t_j, p_i) \in F \\ 0 & \text{en otro caso} \end{cases} \quad a_{ij} = \begin{cases} 1 & \text{si } (t_j, p_i) \in F \\ 0 & \text{en otro caso} \end{cases}$$

La evolución del marcaje M hasta un marcaje M' se establece mediante la siguiente ecuación de estado, la cual es la base algorítmica sobre la que se construye el motor ejecutor de la red de Petri.

$$M' = M + C * u_k \quad (1)$$

Donde u_k es un vector cuyas componentes son nulas, excepto la componente correspondiente a la transición que se dispara en el marcaje M , la cual toma el valor de "1".

Para la representación de un DES mediante PN, los lugares representan estados y las transiciones indican los eventos que producen la evolución dinámica del sistema. Por ser estados binarios, la PN debe estar limitada a un número máximo de una marca en todos los lugares, esto es: $M(p) \leq 1$, ($\forall p \in P$).

Con esta restricción se dice que la PN cumple la propiedad de *acotamiento* (boundedness property) y la propiedad de *seguridad* (safeness property). Además, los modelos para representación del conocimiento mediante PN no deben presentar bloqueos en la ejecución dinámica, por lo que debe cumplir la propiedad de *vivacidad* (liveness property).

IV. MODELOS DE CONOCIMIENTO MEDIANTE PN PARA DIAGNÓSTICO DE FALLAS

Los modelos de conocimiento como representaciones estructuradas hacen uso de símbolos para representar parte del conocimiento y las relaciones entre ellas.

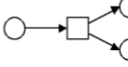
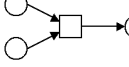


En los procesos de razonamiento que realizan operadores humanos para el diagnóstico de fallas en centrales de generación hidráulica, se presentan estructuras de eventos discretos que pueden ser modeladas mediante PN.

Una PN modela el ciclo de vida de un caso donde las actividades (transiciones) son las partes activas del modelo, mientras que los estados son las partes pasivas (lugares) y los arcos entre transiciones y estados representan relaciones de

causalidad [27]. Dentro de este contexto, una falla se puede modelar como un caso, el cual se activa con la detección de un evento de falla y se termina con el diagnóstico de la misma.

Dentro de los modelos mentales de razonamiento, con características DES, se incluyen relaciones causales, bifurcaciones, precedencias, secuencias, concurrencias e interacciones con otro tipo de sistemas, las cuales pueden ser modeladas con las construcciones AND-split, AND-join, OR-split y OR-join, que se muestran en la Tabla I.

TABLA I. Construcciones de modelamiento

ESTRUCTURA	NOTACIÓN GRÁFICA	ESTRUCTURA	NOTACIÓN GRÁFICA
AND-split		AND-join	
OR-split		OR-join	

La estructura AND-split modela un estado con dos o más transiciones de salida, lo que permite que múltiples actividades sean ejecutadas simultáneamente. La estructura AND-join modela un estado en el que convergen (se sincronizan) dos o más actividades cuya ejecución se puede desarrollar de forma paralela. La siguiente actividad no se inicia hasta que se completa el conjunto de actividades involucradas. La estructura OR-split (se trata de un OR exclusivo), modela el caso en el que dada una determinada actividad, como consecuencia del cumplimiento de determinadas condiciones se inicia una única actividad subsiguiente. La estructura OR-join modela el caso en el que una determinada actividad se inicia, una vez completada una de las actividades precedentes alternativas, no siendo requerida la sincronización de éstas [28].

A partir de las construcciones anteriores se identifican cuatro estructuras de procesamiento para especificar las relaciones entre las tareas durante la ejecución del proceso. [29].

1. Procesamiento secuencial: describe tareas que son ejecutadas secuencialmente, donde cada tarea es seguida por la próxima tarea. La "Fig. 1" ejemplifica un secuencial. La tarea A es ejecutada primero, la tarea B inicia después de que la tarea A es completada y antes de iniciarse la tarea C.

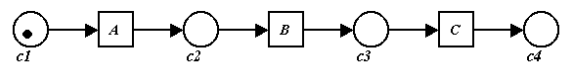


Figura 1. Procesamiento Secuencial

2. Procesamiento paralelo: describe una situación donde dos o más tareas son ejecutadas al mismo tiempo o en el que el orden de ejecución es menos estricto (Ver "Fig. 2). Para

modelar un procesamiento paralelo, se usan dos estructuras: AND-split e AND-join. Se observa también un proceso de sincronización a través de la ejecución de la tarea D.

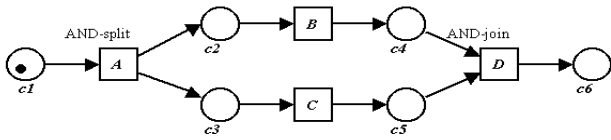


Figura 2. Procesamiento Paralelo

3. Procesamiento condicional: esta construcción es usada para modelar el proceso de elección entre dos o más tareas. Después que la tarea A ha sido completa se elige entre la ejecución de la tarea B o la tarea C. La tarea D se inicia cuando B o C son completadas. Para modelar este tipo de procesamiento se hace uso de dos estructuras: OR-split u OR-join. (Ver “Fig. 3).

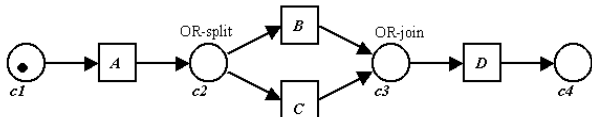


Figura 3. Procesamiento condicional

4. Procesamiento iterativo: representan la ejecución múltiple de una o más tareas. La “Fig. 4” muestra como la tarea B debe ser ejecutada repetidamente hasta que la condición se cumple. La tarea C es una tarea de control, la cual chequea el resultado de la tarea B. Basada en ese chequeo la tarea B puede ser ejecutada más de una vez.

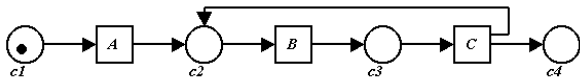


Figura 4. Procesamiento Iterativo

Durante el proceso de diagnóstico de una falla, el operador realiza interacciones con el entorno, las cuales deben ser capturadas por el modelo de conocimiento, para su posterior automatización. Las interacciones se dan con los sistemas de protección y control de la central (relés de protección, controladores lógicos – PLC, unidades terminales remotas – RTU, registradores de secuencias de eventos – SOE); con el sistema de supervisión (SCADA), con sistemas de comunicaciones (radio, celular, mensajes de texto); con bases de datos y con otros sistemas de información (SI).

Para facilitar el proceso de automatización y la representación gráfica de los modelos, las interacciones se representan a través de etiquetas, las cuales permiten a su vez diferenciar las funciones que cada uno de los lugares cumple en la evolución de la red.

En la Tabla II se definen 7 etiquetas y la respectiva interacción que representan:

TABLA II. Etiquetas para representar las interacciones

ETIQUETA	INTERACCIÓN
MOSTRAR REV_ALT_TEM_REC_EXC_26SCR 0	Con el ambiente
SILENCIOSA REV_ALAR 0	Con un Sistema de Información
SEÑAL BANDERA_26SCR 0	Con el ambiente
BOTÓN VEN_REC_NO_EST_PREN 0	Con el operador
INFORMAR ING_VENT_PREND 0	Con el ambiente
FALLA FALL_VENT_O_TRANS_TR1 0	Con el ambiente y con un SI
CONCLUYE CONC_VENT_PREND 0	Con el ambiente y con un SI

Las estructuras de PN presentadas y las respectivas etiquetas para representar las interacciones permiten elaborar modelos de conocimiento con características DES, utilizados por los operadores de centrales de energía para diagnosticar fallas.

En el levantamiento del conocimiento, se establecen las fallas funcionales y para cada falla funcional, los modos de falla que definen las posibles causas. De cada modo de falla se elabora el modelo PN para la automatización del razonamiento empleado para el diagnóstico.

En una central de generación hidráulica se pueden elaborar por ejemplo, los modelos para las fallas de estator y de excitación, con sus respectivos modos de falla. La Tabla III muestra algunos ejemplos de construcciones de conocimiento que hacen uso de las estructuras AND y OR, las estructuras de procesamiento y las etiquetas anteriormente descritas.



Figura 7. Interfaz gráfica - operador



Figura 8. Resultados del diagnóstico

AGRADECIMIENTOS

El contenido de este trabajo, es producto de las actividades de investigación desarrolladas por la empresa EMGESA. S.A. E.S.P y la Universidad Nacional de Colombia - Sede Medellín, en el marco del proyecto de investigación: "DIAGNÓSTICO AUTOMÁTICO DE FALLAS EN CENTRALES DE GENERACIÓN HIDRÁULICA DEL SISTEMA EMGESA", proyecto financiado por COLCIENCIAS. Se agradece a estas entidades por su respaldo en el desarrollo del proyecto.

REFERENCES

- [1] M. Blanke, M. Kinnaert, J. Lunze *et al.*, *Diagnosis and fault-Tolerant control*: Springer Berlin Heidelberg, 2006.
- [2] J. Lunze, "Fault diagnosis of discretely controlled continuous systems by means of discrete-event models," *Discrete Event Dynamic Systems*, vol. 18, no. 2, pp. 181-210, 2008.
- [3] Y. Zhang, and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229-252, 2008.
- [4] Y. Power, and P. A. Bahri, "A two-step supervisory fault diagnosis framework," *Computers & Chemical Engineering*, vol. 28, no. 11, pp. 2131-2140, 2004.
- [5] K. Salimifard and M. Wright. Petri net-based modelling of workflow systems: An overview. *European Journal of Operational Research*, ISSN 0377-2217, 134 (3), 664-676, 2001.
- [6] X. F. Zha, S. Y. E. Lim and S. C. Fok. Integrated knowledge-based Petri net intelligent flexible assembly planning. *Journal of Intelligent Manufacturing*, ISSN 0956-5515, 9 (3), 235-250, 1998.

- [7] S. Dash, and V. Venkatasubramanian, "Challenges in the industrial applications of fault diagnostic systems," *Computers & Chemical Engineering*, vol. 24, no. 2-7, pp. 785-791, 2000.
- [8] H. Ezzedine, C. Kolski, "Modelling of cognitive activity during normal and abnormal situations using Object Petri Nets, application to a supervision system". *Cogn Tech Work*, vol 7, pp 167 – 181, 2005.
- [9] S. Jun, P. Hyun , "Development of automated operating Procedure system using fuzzy colored petri nets for nuclear power plants", *Annals of Nuclear Energy*, vol 31, pp. 849 – 869, 2004.
- [10] J. Pomerol, P. Brezillon, "Operational knowledge representation for practical decision making", *Proceedings of the 34th Hawaii International Conference on System Sciences*, 2001.
- [11] G. Schreiber, B. Welinga , et al., "CommonKADS: A comprehensive methodology for KBS development". *IEEE Expert*, vol 9, n°6, pp. 28 – 37, 1994.
- [12] F. Capkovi, "Modelling and control of discrete event dynamic systems", basis research in computer science. ISSN 0909-0878, 2000.
- [13] J. Jiang, R. Doraiswami, "Design of a real-time knowledge-based controller with applications in hydraulic turbine generator systems". *Journal of intelligent and robotic systems* Vol. 2, pp. 229-244, 1989.
- [14] J. Kurien and M. Moreno. Costs and benefits of model-based diagnosis, ISBN 1095-323X. *IEEE Aerospace Conference*, 1-14, 2008.
- [15] A. Chen and D. Buchs. Towards service-based business process modeling, prototyping and integration. In *Rapid Integration of Software Engineering Techniques*. ISBN 978-3-540-34063-8, Springer Berlin / Heidelberg 3943, 218-233, 2006.
- [16] W. Méndez, "Diseño y construcción de un motor de ejecución de redes de Petri para el análisis de estados de una unidad de producción". Trabajo de grado para optar al título de Ingeniero de Sistemas e Informática. Universidad Nacional de Colombia – Sede Medellín, 2009.
- [17] D. Gutiérrez, "Motor de ejecución de redes de Petri", Universidad de los Andés, Mérida, Venezuela, 2004.
- [18] O.Kummer, F.Wienberg, et. al., "An extensible editor and simulation engine for Petri Nets: Renew", pp. 489 – 493, 2004.
- [19] S. Pelligrini, F. Giacomini, "Design of a Petri Net-Based workflow engine". 3rd International Workshop on Workflow Management and Applications in Grid Environments, pp. 1 - 31. *IEEE Computer Society Washington, DC, USA*, 2008.
- [20] S. Pelligrini, F. Giacomini, et. al, "A practical approach for a workflow management system". In *CoreGRID Workshop on Grid Middleware*, Springer Verlag, Dresden, Germany, 2007.
- [21] R. Eshuis, J. Dehnert, "Reactive Petri Nets for workflow modeling". In: *Proceedings of the 24th International Conference on Applications and Theory of Petri Nets (ICATPN 2003)*, pp 296-315, 2003.
- [22] K.Sarshar, T. Theling,, et al, "Integrating process and organization models of collaborations through object Petri Nets". In: Sarshar, K. (ed) *MKWI 2006*, contribution to XML4BPM, Passau, 2006.
- [23] M.Thandar , M. Dumas, "Business process simulation for operational decision support". *BPM 2007 Workshops*, LNCS 4928, pp. 66-77, 2008.
- [24] C.Masalagiu, W.Chin, et al., "A rigorous methodology for specification and verification of business processes". *Formal Aspects of Computing* , Vol. 21, pp. 495-510, 2009.
- [25] M. Iordache , P. Antsaklis , Supervisory control of concurrent systems: A Petri Net structural approach (*Systems & Control: Foundations & Applications* Birkhauser, 1197724, 2006.
- [26] Akesson, K. "Methods and tools in supervisory control theory". PhD thesis, 2002.
- [27] W. Aalst, "Making work flow: on the application of Petri Nets to Bussiness Process Management". *Lecture Notes in Computer Science* , 23rd International Conference on Applications an Theory of Petri. Vol. 2360, pp. 1 – 22. Adelaide, Australia, 2002.
- [28] P. Solana, M. Martínez, "Análisis y modelado con redes de Workflow del proceso de tratamiento de experiencias operativas".
- [29] K. Grigorova, "Process modelling using Petri Nets", *International Conference on Computer Systems and Technologies* , 2003.