

Las redes de Petri y su adopción en programas de la Universidad de Sonora

Petri nets and their adoption in programs of the university of Sonora

EPISTEMUS

ISSN: 2007-8196 (electrónico)

Eleazar Jimenez Serrano ¹

Recibido: 29 / 01 / 2022

Aceptado: 26 / 09 / 2022

Publicado: 21 / 10 / 2022

DOI: <https://doi.org/10.36790/epistemus.v16i33.221>

Autor de Correspondencia:

Eleazar Jiménez Serrano

Correo: eleazar.jimenez@unison.mx

Resumen

Las redes de Petri (PN) pertenecen a una técnica que está ganando popularidad en diferentes sectores industriales, como el automotriz. En este manuscrito se explican brevemente las PN, su historia y algunas extensiones para aumentar las capacidades en la modelación. Asimismo, dos ejemplos de modelos de PN para la industria automotriz, uno para la verificación de un software para el cálculo de posicionamiento vehicular por GPS, y el segundo para la simulación de una cadena de suministro automotriz. Al final se incluye una breve conclusión orientada a su adopción en los programas del departamento de ingeniería industrial de la Universidad de Sonora (UNISON).

Palabras clave: edes de Petri, simulación de sistemas de manufactura, verificación de software

Abstract

Petri nets (PNs) belong to a technique gaining popularity in different industrial sectors, such as the automotive industry among others. Hereby, PNs are briefly explained, including their history and the description of some extensions useful to increase modeling capabilities. Additionally, two examples of PN models for the automotive industry are presented, one for the software's verification on the calculation of vehicle positioning by GPS, and the second for the simulation of an automotive supply chain. In the end, a brief conclusion is included aiming the adoption of PNs in the programs of the department of industrial engineering of the University of Sonora (UNISON).

Keywords: Petri nets, simulation of manufacturing systems, software verification.

¹ Doctor en Ciencias en Ingeniería de la Computación y Comunicación, Universidad de Sonora, Unidad regional central. División de ingeniería. Correo: eleazar.jimenez@unison.mx, ORCID: <https://orcid.org/0000-0002-2594-6914>

INTRODUCCIÓN

Un tipo de complejidad en la modelación y simulación de sistemas de manufactura está asociado a la ocurrencia de eventos que pudieran ser secuenciales, asíncronos, paralelos y concurrentes. Por ejemplo, la administración de fabricación y distribución de partes y productos en la industria automotriz a nivel mundial contiene este tipo de eventos: manufactura de componentes en fábricas de proveedores en China, México e India, transportación por mar y tierra de las partes a empresas integradoras de sistemas en Corea de Sur, Brasil y España, y finalmente ensamble de los automóviles en lugares como Japón, Estados Unidos y Alemania [1]. Otro ejemplo donde hay estos eventos es en mecanismos de hardware-software de automóviles con tecnología de los sistemas de asistencia avanzada de manejo, conocidos en inglés como Advanced Driver Assistance Systems (ADAS) [2]. El análisis, control y diseño de esta tecnología incluye dispositivos tales como receptores GPS, sensores infrarrojos, radar, cámara y monitor, todos trabajando al mismo tiempo en forma sinérgica.

Las redes de Petri son una técnica relativamente nueva conocidas en inglés como Petri Nets, y llamadas a lo largo de este manuscrito con las siglas PN. Esta técnica está diseñada para modelar y simular sistemas que presentan precisamente eventos secuenciales, asíncronos, paralelos y concurrentes. Tienen su origen a inicios de la década del año 1960, pero están teniendo más popularidad recientemente gracias a los avances tecnológicos en computación.

Este manuscrito presenta las PN como una técnica útil en la simulación y verificación de sistemas. En la siguiente sección se hace una introducción básica de las PN, seguido de aquellas extensiones más útiles y populares. Luego se presentan dos modelos como ejemplos. El primero es del algoritmo computacional para calcular posicionamien-

to vehicular usando señales de GPS. El segundo es de un sistema general de cadenas de suministros en la industria automotriz. Al final se incluye una pequeña sección explicando los beneficios que traería la adopción de PN en los programas de ingeniería industrial de la Universidad de Sonora (UNISON), una universidad que está en un estado con una economía industrial que incluye el sector automotriz.

Redes de Petri

Las PN tienen su origen en una investigación documentada por el doctor alemán Carl A. Petri en 1962 para su trabajo de tesis doctoral en la Universidad de Bonn [3]. Se utilizaron en un principio como un modelo de computación para simular tareas concurrentes en sistemas distribuidos y complejos, un área exclusiva de las ciencias de la computación para el análisis y la verificación de sistemas. Tiempo después ganaron popularidad mundial cuando se usaron con más frecuencia en la década de los noventa para simular otros sistemas más comunes. Por ejemplo, líneas de producción, procesos de servicios bancarios, cadenas de suministros, entre otros.

Las PN tienen un fundamento matemático elegante y sencillo, muy similar a las cadenas de Markov. El tratamiento de la matriz de transiciones de estados está en el dominio de los números enteros. Su representación gráfica explícita permite la visualización del comportamiento dinámico de los resultados de la simulación, algo que los software comerciales nuevos de modelación y simulación están explotando para su popularización.

Es en este siglo veintiuno cuando se materializó el crecimiento exponencial que predijo la ley de Moore respecto al poder computacional. Es también cuando las PN y otros métodos matemáticos que utilizan matrices grandes y requieren mucha memoria se pudieron volver comerciables





en todos los sectores industriales, de servicio, de investigación, educación, etc. Un ejemplo claro es el trabajo realizado en [2] en la industria automotriz.

Redes generales de Petri

La forma general de las PN que presentó el Dr. Petri en 1962 es una representación matemática y gráfica de un sistema a eventos ocurriendo en intervalos discretos en el cual se pueden representar diferentes topologías de sistemas complejos y sus estados, sean estos distribuidos, concurrentes y paralelos.

En su representación gráfica bipartita unidireccional solamente emplea cuatro elementos llamados: condiciones (o places en inglés y descritos con círculos), transiciones (dibujados con rectángulos), arcos y tokens (que son puntos negros adentro de las condiciones). En la figura 1 se identifican estos elementos enumerados según el tipo, algo útil para la identificación gráfica y en la programación para el cómputo.

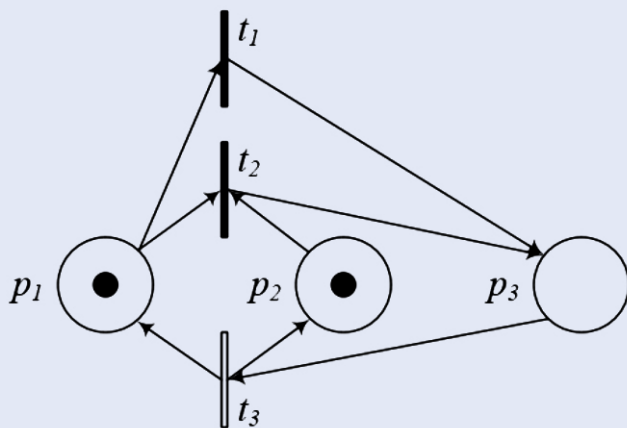


Figura 1. Ejemplo con tres condiciones, tres transiciones y dos tokens.

Las diferentes combinaciones del número de tokens adentro de las condiciones se denominan marcas (markings en inglés), representan los estados de un modelo de PN, y cada marca se representa con un vector m en el dominio de número enteros positivos de tamaño acorde al número de condiciones. Para el ejemplo en la figura 1, la marca actual es $(1, 1, 0)$.

Las reglas para conectar condiciones y transiciones indican usar arcos unidireccionales y no permiten conectar dos elementos iguales. Dichas conexiones se representan con dos matrices en el dominio de número enteros positivos. Para el ejemplo en la figura 1, las dos matrices correspondientes se muestran en los incisos a y b de la figura 2.

a)	b)	c)
t1 t2 t3	t1 t2 t3	t1 t2 t3
p1	p1	p1
p2	p2	p2
p3	p3	p3

Figura 2. Matriz de entrada, de salida, y de transición (resta de b menos a).

Las condiciones conectadas por detrás de las transiciones también se conocen como precondiciones, y las conectadas por delante como postcondiciones. La representación matricial de cómo condiciones y transiciones se conectan permiten calcular la matriz de transición, que para el ejemplo en la figura 1 se muestra en el inciso c de la figura 2.

La fórmula matemática para calcular el cambio de los tokens en las condiciones de la red, la progresión desde una marca actual a una siguiente, es con la siguiente ecuación de transición:

$$m_{siguiente} = m_{actual} + (\text{matriz de transición}) \text{vector}_{\text{transiciones activas por disparar}} \quad (1)$$

La ecuación usa un vector que indica qué transiciones están activas por disparar y ejecuta el comportamiento dinámico. Es de notar que incluye la suma de la marca del estado actual para calcular la marca siguiente. El resultado representa la ejecución de un paso discreto en la simulación.

Las redes de Petri en la configuración general tienen las siguientes características: la capacidad de tokens en las condiciones es de uno, todos los arcos tienen un peso de uno, y la regla para determinar si una transición está activa requiere que cada precondition tenga un token y cada postcondición tenga cero tokens. Todo esto se pone en un algoritmo que se computa de manera secuencial mediante iteraciones en una cantidad de ciclos que depende del tipo simulación, pudiendo ser de exploración exhaustiva o parcial.

En el caso de transiciones activas por disparar en conflicto como el ejemplo en la figura 1, que es dos o más transiciones compartiendo al menos una precondition, la regla de solución puede ser selección aleatoria u alguna otra selección personalizada. Para el caso del ejemplo, si la transición uno fuera la seleccionada a disparar entonces el vector sería (1, 0, 0) y la marca siguiente sería (0, 1, 1).

extensión PRINCIPAL de las redes de Petri

Por el enorme potencial de las PN, desde la década del año 1980 otras capacidades se le han agregado en forma de extensiones. Estas se utilizan para el modelado, análisis, simulación, verificación, control y diseño de sistemas dinámicos, como líneas de producción, procesos de servicios bancarios, cadenas de suministros, etc.

La extensión más común se conoce en inglés como Place/Transitions nets [4]. Esta es la extensión más popular como se conocen las PN y es hoy en día el algoritmo predeterminado. Su configuración es: permite definir una capacidad de más de un token en las condiciones, permite arcos con un peso mayor a uno, y la regla para determinar si una transición está activa se extiende para requerir en cada postcondición que el número actual de tokens más el peso del arco de entrada sea menor o igual que la capacidad de tokens.

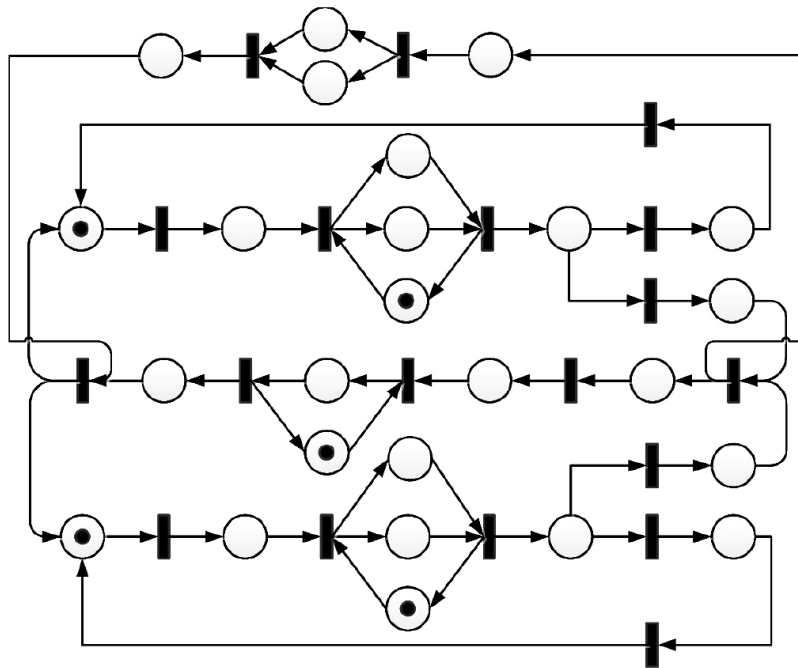


Figura 3. Modelo de red de Petri de un sistema de manufactura flexible.



El ejemplo en la figura 3 proviene de un modelo de un sistema de manufactura flexible presentado en [5]. Se utilizó para simular diferentes escenarios de velocidad de líneas de producción, identificar cuellos de botella y estaciones de trabajo con tiempo muerto, así como estimar el volumen de producción.

Cada una de las 19 transiciones representa una estación de trabajo. Cada precondition es un inventario en la estación de trabajo, el número de tokens es el tamaño del inventario sea de partes o del producto, y el disparo de una transición activa es el término de la operación de producción en la estación. Cada marca del modelo es un vector de 25 componentes (son 25 condiciones) y simboliza un estado en la progresión de la producción en el sistema. La modelación del tiempo de operación de una estación de trabajo se consigue con una extensión de PN con tiempo la cual se presentará a continuación.

Redes de Petri para simulaciones en tiempo continuo

Hay veces que se necesita crear modelos de PN fuera del dominio de pasos a tiempos discretos. Esto para agregar por ejemplo la ocurrencia de las transiciones en el dominio del tiempo continuo, incluir disciplinas prioritarias para dichas ocurrencias, determinar el tiempo de espera de partes o productos en los inventarios bajo disciplinas de salida como la FIFO y la FILO, así como el tiempo que pasa una transición sin disparar. Las siguientes son algunas extensiones de PN útiles para esto.

Redes de Petri CON TIEMPO

El requerimiento de incluir la dependencia del tiempo en el comportamiento dinámico de los modelos al momento de ejecutar la simulación en la actualidad es principalmente para fines de verificación de sistemas que se ejecutan en tiempo real [8], como sistemas automotrices. Sin embargo, esta extensión ha tenido popularidad también en sistemas más sencillos.

Esta extensión se consigue agregando un reloj global a la ejecución, una variable numérica acumulativa en el programa. Luego se crea un vector de tiempo de transición de tamaño al número de transiciones, los valores son unidades de tiempo en términos del reloj. Al algoritmo de PN se le agrega una rutina para ejecutar el comportamiento dinámico siguiendo los tiempos en el vector.

En términos gráficos no existe un elemento que se agregue al modelo de PN. Sin embargo, a nivel de software de aplicación, es común visualizar el reloj global en primer plano durante la simulación.

Redes de Petri ESTOCÁSTICAS

Esta extensión está diseñada para incluir el comportamiento aleatorio de cuándo ocurren las transiciones en el tiempo [9]. Distribuciones de probabilidad, como la exponencial, pueden ser usadas para determinar este tiempo, y los valores son los mismos parámetros usados en la teoría

de líneas: tiempos de llegada, de ejecución del evento, y de salida.

El vector de transición mencionado anteriormente en las PN con tiempo se utiliza aquí de la misma manera, los valores son unidades de tiempo, pero se obtienen de la función de distribución de probabilidad asignada. En términos gráficos no hay componentes nuevos. Los tiempos pudieran agregarse al modelo de PN para ayuda visual exclusivamente.

Ejemplo de red de Petri para la ingeniería de software

Las PN comenzaron como un modelo computacional teórico explicando la teoría de computabilidad usando autómatas abstractos, hasta llegar a lo que se conoce como la teoría de redes de Petri [3]. De allí quizás que la verificación de algoritmos, software y sistemas de cómputo fuera una de las áreas iniciales de desarrollo.

El siguiente ejemplo demuestra el uso de PN en la ingeniería de software. Está tomado del algoritmo general disponible públicamente para calcular el posicionamiento utilizando datos de observación de satélites GPS [2]. El programa está fuera del alcance de este manuscrito, pero en breve se describe de la siguiente forma: utiliza datos decodificados y satelitales para calcular el posicionamiento del receptor, hay suposiciones para categorizar los datos GPS como buenos o malos, y define que hay buenos datos en el caso de tener al menos 4 satélites GPS a la vista. El algoritmo requiere buenos datos por 2 ciclos consecutivos para fijar el posicionamiento y poder extrapolar 1 ciclo en el caso de tener malos datos en el próximo.

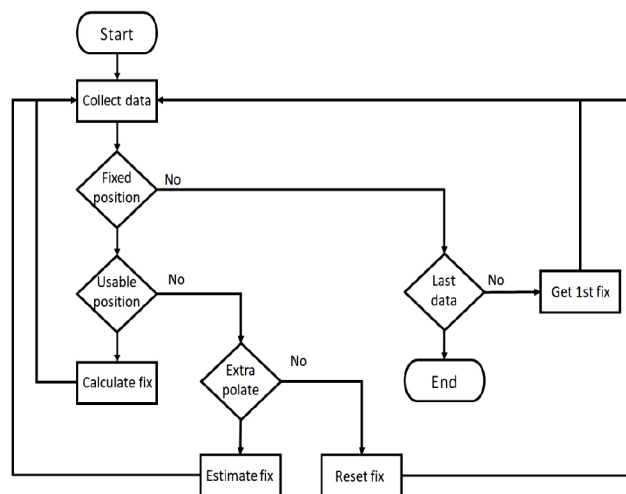


Figura 4. Diagrama de flujo de los estados del sistema de posicionamiento GPS.

Al hacer un modelo de PN, hay diferentes niveles de abstracción del algoritmo en los que nos podemos enfocar. En este manuscrito únicamente se incluirá el nivel de la máquina de estados.

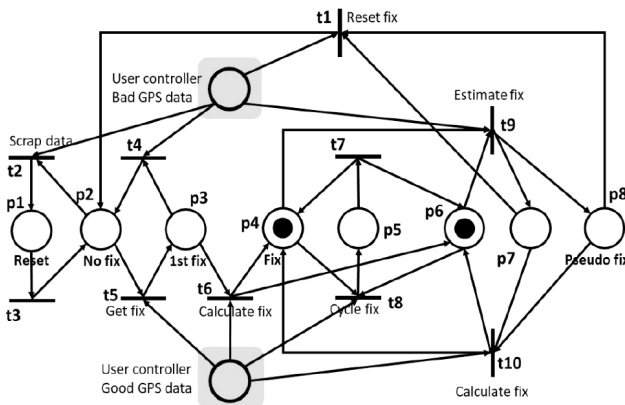
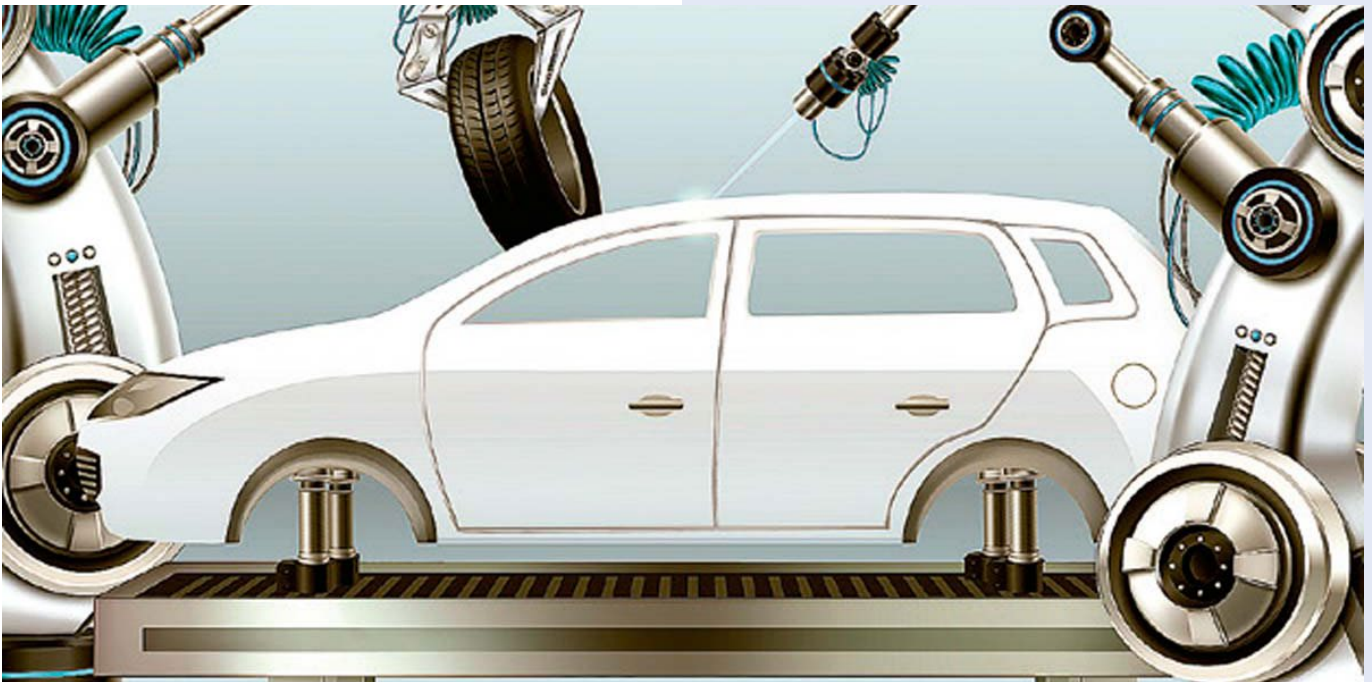


Figura 5. Modelo de red de Petri para el sistema de posicionamiento GPS.

El modelo de PN tiene 10 condiciones de las que 5 representan el estado en el que se encuentra el receptor en el cálculo de la posición. La marca inicial es (0, 1, 0, 0, 0, 0, 0, 0, 0, 0) y correspondería a un token en la condición p2 (No fix). La marca actual en la figura 5 es (0, 0, 0, 1, 0, 1, 0, 0, 0, 0). El control de la progresión de los estados lo determina las dos condiciones que categorizan los datos GPS como buenos o malos. El modelo tiene un comportamiento de exclusión mutua para el caso de determinar la posición cuando se tiene solamente una estimación.

El resultado de una exploración exhaustiva desde la marca inicial bajo una condición constante de datos GPS buenos verifica la progresión del receptor para fijar el posicionamiento mediante el disparo secuencial de las siguientes transiciones: t5, t6, t8, t7.

Ejemplo de red de Petri para la ingeniería industrial
La cantidad de manuscritos académicos se incrementó indudablemente al utilizar la técnica para la simulación y análisis de sistemas dinámicos, muchos de ellos siendo sistemas de producción. En una simple muestra de publicaciones en el portal Academia® disponibles en línea y filtrando solamente aquellos sobre el tema PN, se encuentran casi dos mil manuscritos en los últimos cuarenta años, mostrando un gran crecimiento de este tipo de temas en los primeros años de la década de 1990.

Enfocados en este manuscrito en una cadena de suministro, existe una forma general de crear un modelo de PN. En esta sección, en la figura 6 se muestra el modelo para la cadena de suministro presentada en [10] para la industria automotriz.

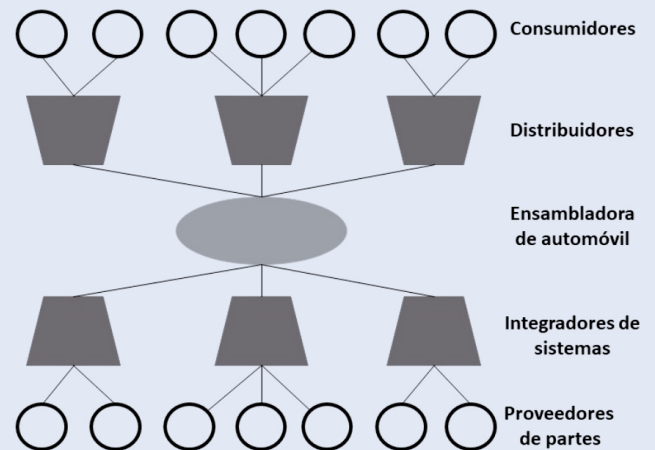


Figura 6. Forma gráfica de una cadena de suministro de la industria automotriz.

Los diferentes actores en la cadena de suministro se representan en un modelo de PN usando condiciones. Los tiempos de transporte, producción y despacho, se presentan en las transiciones como los tiempos de disparo. El comportamiento asíncrono en la simulación se consigue con la extensión de PN estocástica. Las rutas que sigue la cadena de suministro se representan con los arcos que conectan condiciones y transiciones. En las marcas del modelo de PN, el número de tokens representa los inventarios, la cantidad de partes y productos en mano.

La figura 7 muestra un modelo de PN de una cadena de suministros de cinco niveles, comenzando por los proveedores de partes clasificados como Tier-2. La sincronización de la fabricación de los subensambles se observa a nivel de los integradores de sistemas llamados Tier-1. El paralelismo está presente en estos dos niveles. La integración llega a nivel de la ensambladora del automóvil. La sincronización y el paralelismo se observan también en los dos niveles posteriores, en distribuidores y consumidores.

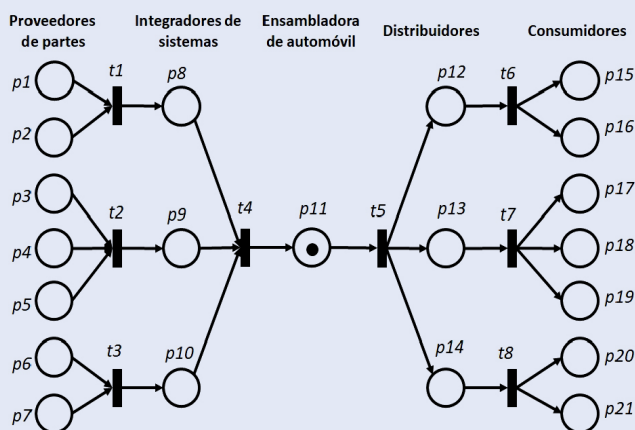


Figura 7. Modelo de PN de la anterior cadena de suministro.

Se nota que para tener un token en la condición p11, esto es tener un automóvil completo y ensamblado, se requirió la participación de todos los proveedores e integradores previos. A nivel de resultado, la suma del número de tokens en las condiciones p15 hasta p21 indican el volumen de producción de automóviles.

CONCLUSIONES

Las redes de Petri desde su inicio han tenido gran impacto en programas de ingeniería de software. Por ejemplo, ya al final de la década del año 1980 el número de publicaciones explicando el uso de PN en la verificación de propiedades de software era alto, y el resumen en [11] lo demuestra. En asignaturas de diseño de software, ya en este siglo, un trabajo como en [12] muestra cómo se pueden utilizar las PN para mejorar la representación de un sistema ya especificado con diagramas elaborados con Unified Modeling Language (UML).

En el campo de la ingeniería industrial, las redes de Petri han ocupado un lugar importante al permitir modelar y simular líneas de producción, cadenas de suministro, y hasta sistemas en la complejidad del ámbito de Industria 4.0. En muchos programas de ingeniería industrial de universidades extranjeras, las PN están como parte fundamental de las asignaturas sobre investigación de operaciones, toma de decisiones y simulación de sistemas.

En general, la adopción de PN en los programas del departamento de ingeniería industrial de la UNISON apuntaría hacia un alto grado de actualización académica en sus contenidos, a una internacionalización que permitiría a su cuerpo académico abordar temas de investigación de mayor novedad e interés en universidades extranjeras, y a ofrecer una ventaja competitiva a sus egresados al buscar colocarse en sectores industriales, como el automotriz.

BIBLIOGRAFÍA

- [1] Eleazar Jiménez Serrano. Reporte de trabajo interno a Nidec Corp. sobre la cadena de suministro a nivel Tier-2 de motores eléctricos de la unidad de negocios automotrices de Nidec. Shiga, Japón. Noviembre 2019, "sin publicar".
- [2] Eleazar Jiménez Serrano. System and software verification using Petri nets. For functional safety of ISO-26262 requirements. Ed. New York, NY, USA: Amazon press, 2020.
- [3] Wolfgang Reisig. Understanding Petri nets. Springer-Verlag Berlin Heidelberg 2013. DOI 10.1007/978-3-642-33278-4.
- [4] W. Brauer, W. Reisig, G. Rozenberg. Petri nets: Central models and their properties. Ed. Springer-Verlag. 1987. DOI 10.1007/978-3-540-47919-2
- [5] Eleazar Jiménez Serrano. "Estimation of the size of state space of Petri nets to determine the size of hash tables," 12th WSEAS Int. Conf. on Applications of Computer Engineering, 2013, pp. 55-60.
- [6] N. Busi. Analysis issues in Petri nets with inhibitor arcs. Theoretical Computer Science No. 275, pp. 127-177, 2002.
- [7] Eleazar Jiménez Serrano. Multiple and simultaneous control using Controlled Dan Petri nets. Tesis de doctorado. Universidad de Kyushu. Marzo, 2008.
- [8] Louchka Popova-Zeugmann. Time and Petri Nets. Ed. Springer. 2013. DOI 10.1007/978-3-642-41115-1.
- [9] G. Balbo. Introduction to generalized stochastic Petri nets. Springer LNCS 4486, pp. 83-131, 2007.
- [10] H. Van Dyke Parunak, Robert Savit, Rick L. Riolo. "Agent-based modeling vs. equation-based modeling: a case study and users' guide," Proceedings of Multi-agent systems and Agent-based Simulation, 1998, pp. 10-25.
- [11] T. Murata. "Petri Nets: Properties, analysis and applications," Proceedings of the IEEE, Vol. 77, No.4, April 1989, pp. 541-580.
- [12] B. Bordbar, L. Giacomini, D.J. Holding. "UML and Petri nets for design and analysis of distributed systems," Proceedings of the IEEE International Conference on Control Applications, 2000, pp. 610-615.

Cómo citar este artículo:

Jimenez Serrano, E. (2022). Las redes de Petri y su adopción en programas de la UNISON. EPISTEMUS, 16(33). <https://doi.org/10.36790/epistemus.v16i33.221>