

Simulación de un sistema de manufactura flexible con redes de Petri coloreadas

Fecha de recepción: 14/04/2009

Fecha de aceptación: 20/06/2009

Luis Diego Murillo Soto¹

Palabras clave

Sistema de Manufactura Flexible (SMF),
Redes de Petri Coloreadas (RdPC),
CPNTool, Simulación.

Key words

Flexible manufacturing systems (FMS),
Coloured Petri Nets (CPN), CPNTool,
Simulation.

Resumen

Una forma de estudiar un sistema de manufactura flexible es por medio de un modelo del sistema y someterlo a un simulacro para observar sus índices de comportamiento, gráficas, etc. Siguiendo esta pauta, el presente trabajo plantea la simulación de una línea de producción con tres máquinas, dos robots y tres tipos de productos distintos. Con dicho sistema se desea obtener, en forma automática, porcentajes del uso de las máquinas, tamaño promedio de las colas, gráficas del comportamiento de las colas y el diagrama Gantt. El modelado de la línea de producción se realizó con Redes de Petri

Coloreadas (RdPC) y su implementación se realizó con la herramienta CPNTool.

Abstract

One way to study a flexible manufacturing system is making a model of the system and simulates it, to observe its behavior indices, graphs, etc. Following this guideline, the present work plans, the simulation of a production line with three machines, two robots and three types of different products. For this system is desired to obtain machines utilization, queues average size, charts of queues behavior and the automatic generation of the diagram Gantt. The modeled one of production line was made with Coloured Petri Nets (CPN) and its implementation was made with the CPNTool tool.

Introducción

Este artículo describe los resultados de un proyecto² académico realizado a finales del 2008, cuyos objetivos específicos fueron los siguientes:

1. Profesor de la Escuela de Ingeniería Electromecánica del Instituto Tecnológico de Costa Rica.
Correo electrónico: lmurillo@itcr.ac.cr
2. Proyecto desarrollado en el Seminario II de la Maestría en Ingeniería Eléctrica de la UCR.

- Investigar fuentes bibliográficas que relacione las RdPC y la manufactura.
- Seleccionar una herramienta computacional de uso libre que permita construir, editar y simular modelos con RdPC.
- Dominar la herramienta seleccionada.
- Modelar y simular una celda de manufactura flexible.

Este documento pretende mostrar hechos importantes del proceso de investigación por lo que se dividirá en seis secciones. La sección II plantea el problema a resolver, los requerimientos y los supuestos. La sección III presenta los principales criterios utilizados para la selección de la herramienta, así como una breve introducción de la misma. La sección IV consiste en la definición matemática de las Redes de Petri Coloreadas con la finalidad de conocer cada componente y así eliminar ambigüedades cuando se presente el modelo. En la sección V se presenta el modelo del proceso realizado con la herramienta CPNTool, así como una explicación de cada parte de la RdPC. La sección VI muestra los resultados de la simulación, gráficas e índices. Finalmente, en la última sección

se brindan recomendaciones para futuros trabajos.

Planteamiento del problema y sus supuestos

El sistema de manufactura que se desea modelar cuenta con tres máquinas, dos brazos robot de siete grados de libertad y seis almacenes de materiales que se encuentran a la entrada y salida de cada máquina. En total, la línea de producción trabaja tres tipos de producto o materiales, los cuales llamaremos “A”, “B” y “C”. La figura 1 muestra un esquema del proceso productivo. Para este proceso de manufactura, se desea construir un modelo con RdPC y hacer una simulación con el fin de:

- Generar automáticamente el diagrama de GANTT, que muestre el recorrido de las piezas a lo largo de todo el proceso.
- Generar porcentajes de utilización de cada máquina y de cada robot.
- Generar los promedios ponderados de piezas en cada cola.
- Generar el conjunto de datos para construir las gráficas de material en cola.

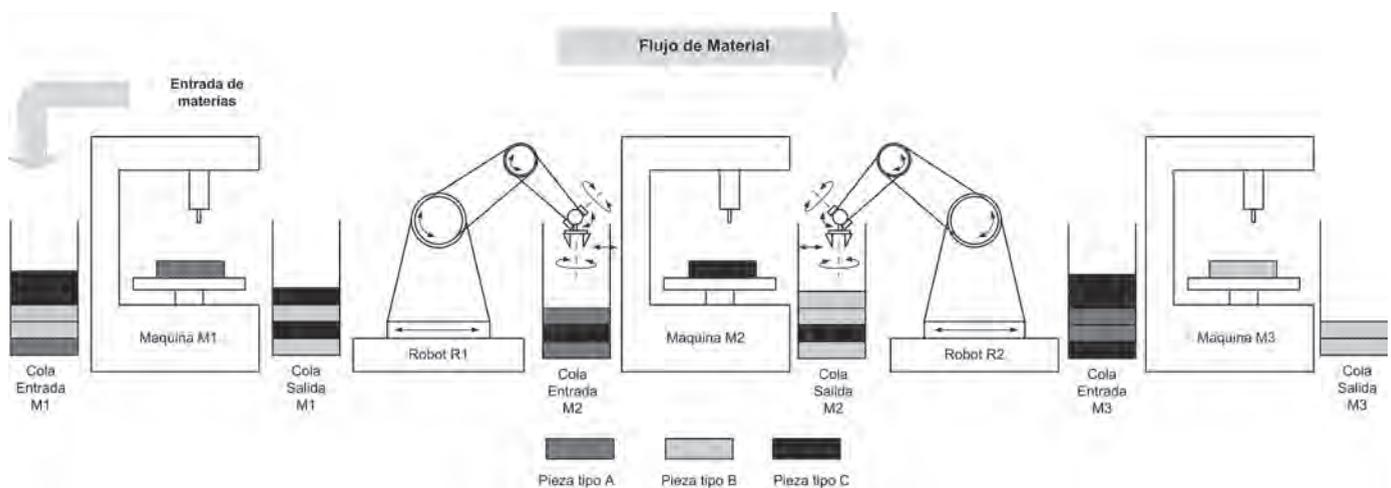


Figura 1. Esquema del proceso productivo que se desea modelar. Se observa tres tipos de productos, las colas de espera, tres máquinas y dos robots.

Dado que este es un problema teórico que no responde a ningún proceso de manufactura existente, se deberá suponer la tasa de llegadas de nuevos objetos λ , así como las duraciones de los objetos cuando una determinada máquina o robot los esté manipulando. Las duraciones de los tiempos de manipulación de cada máquina se presume que poseen un comportamiento estadístico, por lo que se utiliza la función de distribución normal, $N(\mu, v)$ con μ como la media y v como la varianza, lo cual permitirá que el modelo tenga un comportamiento “real”. Por ejemplo, la tasa de llegadas λ se

especificó en $N(10,4)$, cuyo significado es que, en promedio, cada diez unidades de tiempo arriba a la celda una pieza del tipo A, B, o C; además la curva normal presenta una varianza de cuatro, que la hace más abierta o achatada. La selección de cuál pieza ingresa al sistema se realiza aleatoriamente (distribución uniforme). El cuadro 1 asigna las duraciones para cada máquina del sistema según el tipo de pieza.

Para el modelo del robot de siete grados de libertad, se utilizó como referencia el ARMDROID 2001 de Lab Volt. Si se estudia su operación, se determina que una forma de modelar el robot es mediante los puntos de referencia que se programan para recoger, trasladar, entregar una pieza y devolverse. De esta manera se determina que el robot posee seis puntos de referencia y los recorre en ocho movimientos distintos. El cuadro 2 muestra las duraciones de cada movimiento, según el tipo de pieza para ambos brazos robots.

Cuadro 1. Asignación de duraciones

Material	Máquina 1	Máquina 2	Máquina 3
A	$N(20,1)$	$N(10,1)$	$N(5,1)$
B	$N(10,1)$	$N(15,1)$	$N(20,1)$
C	$N(15,1)$	$N(20,1)$	$N(10,1)$

Cuadro 2. Duraciones para cada movimiento de los robots, según producto a manipular.

Movimiento del robot	Tipo de Producto					
	A		B		C	
	Robot 1	Robot 2	Robot 1	Robot 2	Robot 1	Robot 2
Del Punto 1 al Punto 2: (P1 \square P2) Baja y abre la pinza	$N(2,1)$	$N(2,1)$	$N(2,1)$	$N(2,1)$	$N(2,1)$	$N(2,1)$
Del Punto 2 al Punto 3: (P2 \square P3) Cierra la pinza (toma objeto)	$N(1,1)$	$N(1,1)$	$N(1,1)$	$N(1,1)$	$N(1,1)$	$N(1,1)$
Del Punto 3 al Punto 1: (P3 \square P1) Mueve arriba brazo	$N(2,1)$	$N(3,1)$	$N(3,1)$	$N(4,1)$	$N(2,1)$	$N(3,1)$
Del Punto 1 al Punto 4: (P1 \square P4) Traslado de la pieza	$N(5,1)$	$N(5,1)$	$N(6,1)$	$N(6,1)$	$N(4,1)$	$N(4,1)$
Del Punto 4 al Punto 5: (P4 \square P5) Baja la pinza	$N(2,1)$	$N(3,1)$	$N(3,1)$	$N(4,1)$	$N(2,1)$	$N(3,1)$
Del Punto 5 al Punto 6: (P5 \square P6) Abre pinza (entrega objeto)	$N(1,1)$	$N(1,1)$	$N(1,1)$	$N(1,1)$	$N(1,1)$	$N(1,1)$
Del Punto 6 al Punto 4: (P6 \square P4) Sube el brazo	$N(2,1)$	$N(2,1)$	$N(2,1)$	$N(2,1)$	$N(2,1)$	$N(2,1)$
Del Punto 4 al Punto 1: (P4 \square P1) Retorno del brazo	$N(3,1)$	$N(3,1)$	$N(3,1)$	$N(3,1)$	$N(3,1)$	$N(3,1)$

Búsqueda y selección de la herramienta

La búsqueda de un software adecuado para editar y simular RdPC es un proceso que consume gran cantidad de tiempo, debido a la disponibilidad de numerosos paquetes en la *World Wide Web*. Por ejemplo, la página “*Petri Nets World*” de la Universidad de Hamburgo [11], registra más de setenta herramientas de evaluación/simulación y cerca de dieciséis de ellas corresponden a herramientas para trabajar con Redes de Petri Coloreadas. La selección formal de un software implica que, con anterioridad, se debe definir una serie de requerimientos de evaluación tales como: desempeño, tamaño, fiabilidad, robustez, portabilidad, usabilidad, etc., sin embargo esto no es factible de realizar, por lo que se recurrirá al criterio de expertos y a los siguientes requerimientos:

- El software debe ser de licenciamiento libre y preferiblemente que se utilice en entornos académicos.
- El software debe respetar la sintaxis y la semántica de las RdPC, tal y como se propone en [4,5].
- El software debe ser un editor gráfico, con características de simulación y reporte.
- El software debe ser compatible con el sistema operativo Windows XP.
- El software debe poseer soporte en línea, vía correo y amplia documentación.

En 1998 la Universidad de Munich realizó un estudio comparativo de todos los softwares existentes para el análisis y simulación de Redes de Petri de alto nivel. Ese estudio [9] incluyó 91 paquetes y determinó entre los paquetes tanto comerciales como académicos, el que mejor puntaje obtuvo a través de todas las pruebas realizadas fue el llamado Design/CPN. Más recientemente, en el 2007, Hrúz [3] recomienda la herramienta CPNTool. Este paquete es el sucesor del Design/CPN

y ambos softwares fueron desarrollados por la Universidad de Aarhus, bajo la guía de Kurt Jensen, proponente de las Redes de Petri Coloreadas. Dado que CPNTool cumple con todos los requerimientos planteados, se escogió esta herramienta para su aprendizaje, así como para la construcción y simulación del sistema de manufactura.

CPNTool y otras herramientas

La herramienta CPNTool es un software especializado en la declaración, simulación y análisis de Redes de Petri Coloreadas Jerarquizadas Temporizadas. Esta sección presenta los componentes más importantes de la interfase gráfica sin pretender ser un tutorial. Existe literatura especializada [7,12] que documenta en detalle todas las partes del paquete y su óptima utilización en la creación, simulación y análisis de una RdPC. Este software está disponible en forma gratuita, así como toda su documentación en [10].

La interfase gráfica de usuario (GUI) de la herramienta CPNTool, no posee menús desplegables, solo usa un “*scrollbar*” y muy pocos cuadros de diálogo. Los dos componentes principales del GUI son el *Índice* y el *Espacio de trabajo*. El Índice es una columna que enlista todos las herramientas disponibles y es similar al menú tipo árbol del administrador de archivos de “*Windows Explorer*”. El índice, consta de cuatro entradas principales que aparecen en el siguiente orden: “*Tool box*”, “*Help*”, “*Options*”, y los proyectos cargados en la herramienta (ver figura 2).

La entrada “*Help*” brinda acceso a la ayuda del programa, mientras que la entrada “*Options*” muestra cada cuántos comandos se auto-salvarán los archivos generados por la GUI. La entrada “*Tool box*” despliega todos los grupos de herramientas disponibles, llamados paletas; la manera de desplegar las paletas en pantalla, es al seleccionar la paleta buscada y arrastrarla al espacio de trabajo. Las paletas existentes

son: “Auxiliary”, “Create”, “Hierarchy”, “Net”, “Simulation”, “Monitoring”, “State Space”, “Style” y “View”. La paleta “Auxiliary” agrupa las herramientas para crear etiquetas informativas, estas herramientas no aportan significado semántico a la red. La paleta “Create”, contiene herramientas, tales como creación de transiciones, lugares y arcos dirigidos, con lo que se dibuja la topología de una red. La paleta “Hierarchy” posee las herramientas destinadas a la creación de jerarquías entre redes. Por otro lado, la paleta “Net” se utiliza para salvar y cargar redes al GUI, así como para la exportación de las redes al formato de gráfico “Encapsulated PostScript” (EPS). Las paletas “Simulation” y “Monitoring” se encargan de ejecutar la red y capturar datos de la evolución de las marcas a través de la red. La paleta “State Space” posee un conjunto de herramientas para el análisis del espacio de estados de la red y la generación automática del grafo de estados. Finalmente, la paleta “Style” posee herramientas para el color, ancho, estilo de líneas de texto, mientras que “View” contiene herramientas para realizar vistas parciales, acercamientos y desplazamientos de la red. En la figura 2 se aprecia parte de las herramientas descritas anteriormente.

El proyecto cargado en el sistema es visto en el índice por su nombre, la figura 2 muestra el nombre del proyecto *SistemaManufactura.cpn*. La GUI puede soportar varios proyectos cargados de forma simultánea, y para cada proyecto se cargan en el índice las siguientes entradas: “Step”, “Time”, “Options”, “History”, “Declarations” y “Monitors”. La entrada “Step” contabiliza la cantidad de pasos ejecutados en la red cuando se realiza una simulación, es decir, este parámetro indica la cantidad de transiciones disparadas, mientras que la entrada “Time” muestra el tiempo simulado actual de la red. La entrada “Options” permite configurar la dirección de las carpetas donde se salva el archivo y los reportes, etc. Por

último “History” muestra un recuento de comandos ejecutados en la red y la entrada “Monitors” es un listado de herramientas de captura de datos que se aplicaron a la red. Finalmente, la entrada “Declarations” es donde se definen los tipos de datos a manejar por la red (colores), las variables y las funciones; la sintaxis de las declaraciones para nuevos tipos de datos, variables y funciones, se realiza de acuerdo al lenguaje CPN ML que es propio de CPNTool. Para la utilización adecuada de CPN ML es necesario conocer el lenguaje “Standar ML”, cuya documentación esta disponible en [1].

El espacio de trabajo es un área color gris a la derecha de la GUI, en ese espacio se cargan ventanas flotantes llamadas “binder”, que a su vez contienen una o varias páginas. Sobre cada página es posible crear y editar una RdPC o parte de ella. La forma en que se crean los “binder”, las páginas contenidas en los “binder”, las redes dentro de las páginas, etc, es por medio de un menú contextual llamado menú de marcas, que es fácil de reconocer por ser circular. Este menú aparece con el botón derecho del ratón y las opciones del menú dependen de la zona específica en la GUI que se seleccione. Así, por ejemplo, en la figura 2, se pulsó sobre la página llamada Robot 2, y el menú de marcas ofreció las siguientes opciones de acuerdo a ese contexto: creación de lugares, transiciones, etc.

El software CPNTool presenta herramientas de monitoreo de la red que permiten exportar los datos capturados y guardarlos en archivos de texto. Los archivos salvados luego se pueden procesar en hojas de cálculo u otros programas, sin embargo, a diferencia del software predecesor, CPNTool no posee herramientas para la generación automática de gráficos, por lo que hay recurrir al software llamado BRITNeY. Dicho paquete informático esta disponible en forma gratuita en [14], BRITNeY se destina a la creación de más

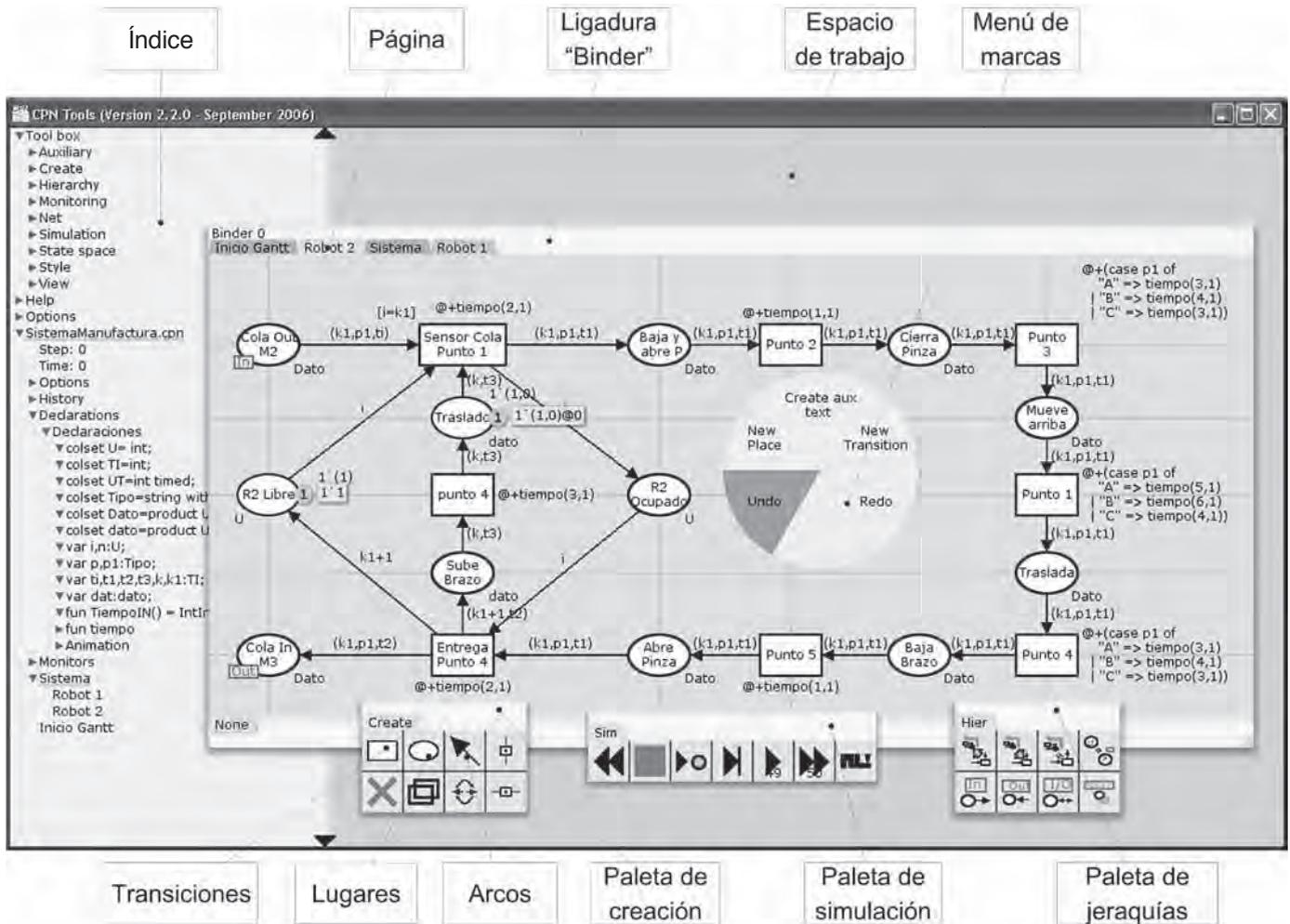


Figura 2. Interfase gráfica de usuario de CPNTool. Se aprecia los componentes principales: índice, espacio de trabajo, ligaduras, páginas, paletas de herramientas, herramientas particulares y el menú contextual (marcas). La RdPC que se observa corresponde al modelo del robot dos.

de quince tipos de gráficas (gant, pastel, barras, etc.) y animaciones, la explicación de cómo crear las gráficas y establecer el enlace con CPNTool está disponible en el artículo [13], que es un tutorial paso a paso para aprender a usar la herramienta.

Redes de Petri Coloreadas

El modelo realizado del proceso productivo corresponde a una RdPC jerarquizada y temporizada, lo cual es una extensión de la RdPC original definida en 1981 por Jensen [2,3,4,5]. Sin embargo, para lograr entender el modelo, es necesario

hacer una introducción de las Redes de Petri Coloreadas. Si se desea una mayor comprensión y profundidad del tema se puede consultar [4,5].

Las RdPC utilizan diversos conceptos y definiciones. A continuación se presenta algunas definiciones importantes para el entendimiento posterior de la definición de RdPC:

- Los elementos de tipo s , son el conjunto de elementos contenidos en S .
- El tipo de una variable v , se denota por $Type(v)$.

- El tipo de una expresión exp , se denota por $Type(exp)$.
- El conjunto de variables de una expresión exp , se denota por $Var(exp)$.
- La letra \mathbb{B} denota el conjunto de elementos tipo booleano {verdadero, falso}.
- Las RdPC utilizan el concepto de multi-conjunto para explicar qué pasa con los colores (datos) en un nodo. Se podría definir un multi-conjunto como un conjunto, pero con la diferencia de que pueden existir múltiples apariciones del mismo elemento. Para ejemplificar, suponga que tiene un conjunto llamado S con los elementos $\{a,b,c\}$, si a este conjunto le agrega el elemento b , lo que obtienen es el mismo conjunto S . Ahora bien, si el conjunto S se define como multi-conjunto, entonces existen dos elementos de b , por lo que el multi-conjunto es $\{a,b,b,c\}$. Cada nueva aparición de un elemento s existente en S , genera un nuevo multi-conjunto, así si se agrega nuevamente b , el nuevo multi-conjunto es $\{a,b,b,b,c\}$. De esta forma, es posible afirmar que existe una infinita cantidad de multi-conjuntos definidos sobre S dado que se pueden añadir n elementos de a , b o c . La manera de representar el conjunto de todos los multi-conjuntos sobre S , es S_{MS} .

En [4] se define una Red de Petri Coloreada no jerarquizada como una tupla $RdPC=(\Sigma,P,T,A,N,C,G,E,I)$, donde :

- Σ es un conjunto finito no vacío de tipos de colores, llamado **colección de colores**. Los colores permiten especificar los atributos de cada entidad u objeto que se desea modelar, es decir, se definen los tipos de datos con que se trabajará en el modelo.
- P es un conjunto finito de nodos llamados **lugares**.
- T es un conjunto finito de nodos

llamados **transiciones**. Las transiciones se asociarán al conjunto de eventos discretos que suelen suceder al inicio o final de una serie de actividades.

- A es un conjunto finito de arcos dirigidos, tal que $P \cap T = P \cap A = T \cap A = \emptyset$.
- N es una función de nodo, $N(a)$, que permite asociar a cada arco sus nodos terminales (origen, destino). Los nodos terminales deben ser de tipo distinto, por tanto, si el nodo origen es una transición, el otro debe ser un lugar, o viceversa. Es decir, N define A dentro $P \times T \cup T \times P$.
- C es un conjunto de funciones de color $C(p)$, que especifica, para cada nodo lugar, el tipo de entidades que puede almacenar. Esto significa que cada marca sobre p debe poseer un color que pertenezca al tipo $C(p)$.
- G es una función llamada **guarda**, la cual se asocia a los nodos tipo transición $G(t)$, esta función puede especificar una lista de expresiones booleanas que cuando son verdaderas, habilitan la respectiva transición. La definición es,

$$\forall t \in T : [Type(G(t)) = \mathbb{B} \wedge Type(Var(G(t))) \subseteq \Sigma]$$

E es la función que asocia las expresiones a los arcos, $E(a)$. Las expresiones deben ser del tipo $C(p(a))_{MS}$, esto significa que cada evaluación de la expresión del arco debe arrojar un multi-conjunto definido sobre la colección de colores que está asociada al lugar respectivo. Esto es:

$$\forall t \in T : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma]$$

Es decir, las expresiones $E(a)$ ligadas al arco de entrada de una transición, especifican el tipo de objeto del nodo lugar de entrada que debe seleccionarse

de entre los objetos almacenados en ese nodo lugar, para habilitar la transición al momento del evento. Si la expresión $E(a)$ está vinculada al arco de salida de una transición, la expresión se utiliza para evaluar los nuevos valores de los atributos de los objetos de salida.

- I es la función de inicialización y permite determinar los atributos de los objetos (colores de las marcas) que se almacenan desde un principio en los lugares. Es decir, el estado inicial del sistema, conocido como marcado inicial M_0 , que es el conjunto de objetos y sus colores almacenados inicialmente en los nodos lugar, se especifica con la función I . La función I cumple que:

$$\forall p \in P: [Type(I(p)) = C(p)_{MS}]$$

Las funciones de inicialización son del tipo multi conjunto y se especifican de la forma

$$n_1'(c_1, c_1, \dots, c_k) + n_2'(c_1, c_1, \dots, c_k) + \dots + n_i'(c_1, c_1, \dots, c_k)$$

Donde:

n_i = número de objetos que poseen los valores de colores descritos entre paréntesis.

c_i = valor del color, que puede ser desde ningún valor hasta el último. No se pueden repetir valores de color en la misma asociación.

Debido a que se desea obtener métricas de rendimiento para el sistema productivo planteado, es necesario definir una Red de Petri Coloreada Temporizada (RdPCT), la cual se entiende como una tupla $RdPCT = (RdPC, R, r_0)$ tal que:

- RdPC es la red antes definida con el siguiente cambio, tanto en $E(a)$ como $I(p)$ pueden, o no, ser funciones

temporizadas sobre $C(p(a))$ y $C(p)$.

- R es un conjunto de valores de tiempo, llamados **estampas de tiempos**.
- r_0 es un elemento de R , llamado tiempo inicial.

Las definiciones anteriores establecen cómo es la estructura de una RdPC o una RdPCT, sin embargo, no especifican el comportamiento dinámico; es decir, no definen el conjunto de reglas que rigen la evolución de los objetos dentro de la red. Para una explicación detallada y formal ver [4], mientras que [6] es una introducción narrativa basada en ejemplos, que evita las definiciones matemáticas.

Modelado del proceso

El sistema productivo descrito en una sección anterior se clasifica como un sistema de eventos discretos, según [8]. Existen diversos formalismos para el modelo de este tipo de sistemas, sin embargo, las RdPC representan, en forma natural, sistemas de este tipo. Para el sistema productivo planteado, cada evento del sistema se asociará a un nodo transición y cada actividad se asociará a un nodo lugar de la red, así una transición disparada provoca un cambio en las variables de estado del sistema, y cada lugar de la red encapsula lo que sucede entre dos eventos. De acuerdo con lo anterior, el estado del sistema de producción estará determinado por todas las marcas distribuidas en los nodos lugares. Esto significa que cada evento del sistema, conceptualizado como la finalización de una actividad y principio de otra, se representará como un nodo transición de la RdPC. Las actividades del sistema, vistas como la manipulación o transporte de piezas, la espera en cola de un producto por falta de recursos, el movimiento de un punto a otro del robot, etc., se representarán como nodos lugar. En la herramienta CPNTool, los lugares de una red se representan como óvalos,

las transiciones como rectángulos y los arcos dirigidos como flechas, los arcos dirigidos en ambos sentidos como una flecha bidireccional.

El modelo del sistema fue elaborado con el método constructivo “Top-Botton”, de lo general a lo específico. El nivel superior de la red representa las máquinas, los almacenes de espera (colas) y los arribos de las piezas; en un nivel inferior se modelan los dos robots del sistema. La figura 3 muestra la RdPC del nivel superior llamada Sistema: en color rojo se modela el arribo de piezas al sistema, en color azul

se modela la máquina con sus colas de entrada y salida, en color verde se modela la máquina 2 y sus respectivas colas, y en color café se modela la máquina 3 con sus colas. En este nivel los robots no son modelados, pero se señalan como una transición, que se representa en la figura como un doble rectángulo. La figura 4 muestra la RdPC para el robot 1; la figura para el robot 2 no aparece debido a su similitud topológica con el robot

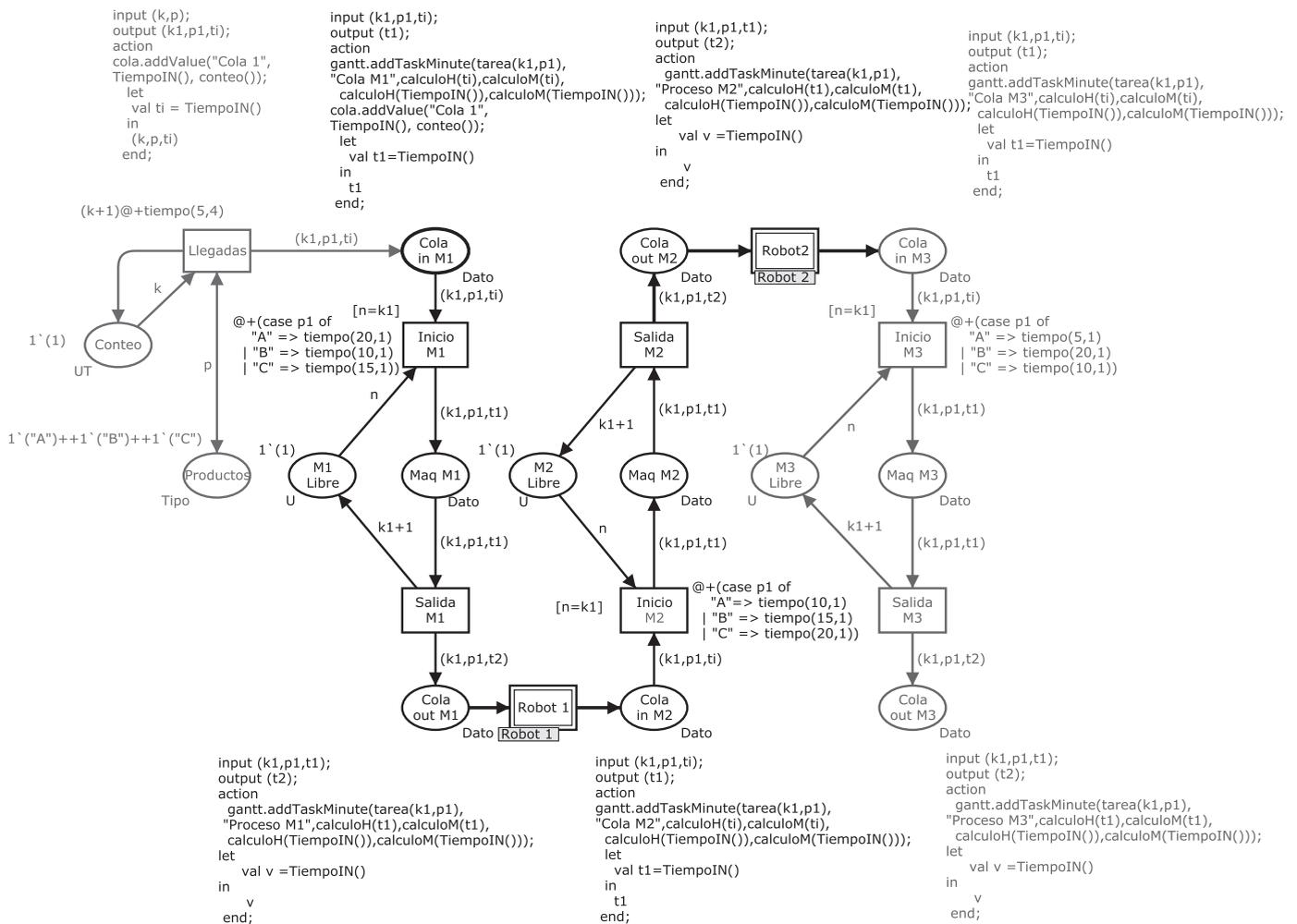


Figura 3. Representación del sistema productivo por medio de una red coloreada, temporizada y jerarquizada. Las transiciones con doble rectángulo indican que existe, en una RdPC de nivel inferior, una para cada brazo robot.

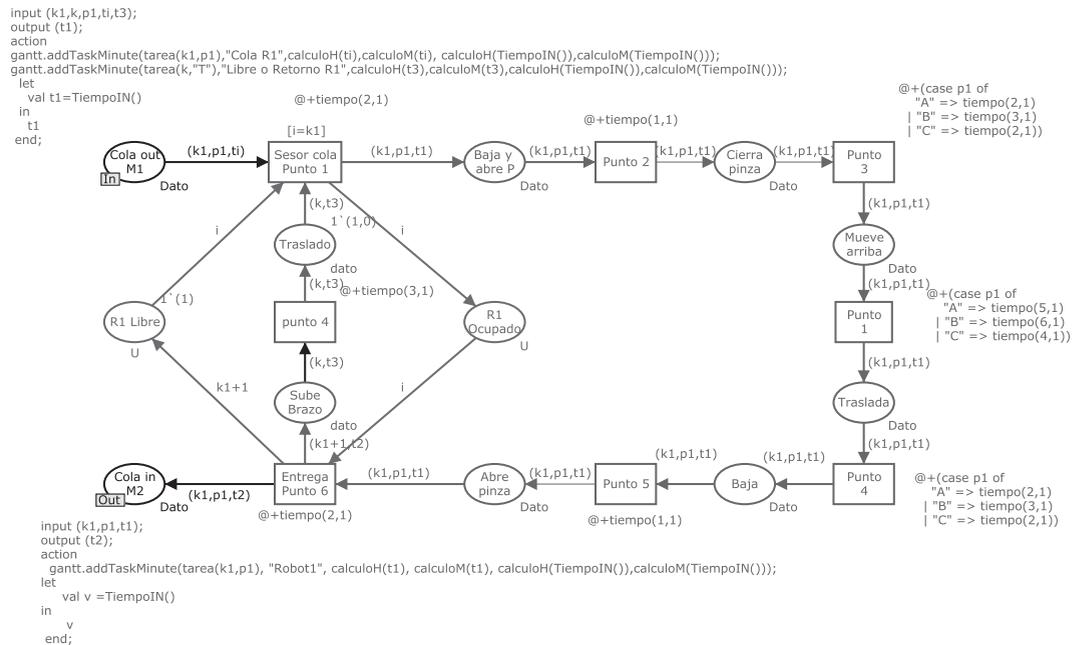


Figura 4. Modelo del Robot 1. El nodo lugar Cola out M1 funciona como un puerto de entrada y el nodo lugar Cola in M2 funciona como un puerto de salida de la red de nivel inferior. Los nodos lugares y arcos de color rojo modelan si el robot esta ocupado transportando una pieza, o está libre esperando una pieza de la cola de salida de M1. La sección de la red en color violeta modela el transporte de la pieza, el retorno del brazo robot a su punto inicial y su duración en cada movimiento.

Cuadro 3. Declaraciones de los colores, variables y funciones según lenguaje CPN ML.

Colores	Funciones
<pre> colset U= int; colset TI=int; colset UT=int timed; colset Tipo=string with "A".."C"; colset Dato=product U*Tipo*TI timed; colset dato=product U*TI timed; </pre>	<pre> fun TiempoIN() = IntInf.toInt (time()); fun tiempo(promedio:int, varianza:int)= let val realPrd = Real.fromInt promedio val realvr = Real.fromInt varianza val rv = normal(realPrd,realvr) in floor (rv+0.5) end; structure gantt=GanttChart(val name="Prueba"); structure cola=XYChart(val name="Prueba2"); fun tarea(ume:int, producto:string)= let val s1=Int.toString ume val s2=s1^producto in s2 end; fun calculoH(tin:U)= let val tmp=tin; val m= 60; val x=tmp div m; in x end; fun calculoM(tin:U)= let val tmp=tin; val m= 60; val y1= tmp mod m; in y1 end; </pre>
Variables	
<pre> var i,n:U; var p,p1:Tipo; var ti,t1,t2,t3,k,k1:TI; var dat:dato; </pre>	

1. Para interpretar correctamente estas RdPC, es necesario ver las declaraciones de colores (datos), variables y funciones que se muestran del cuadro 3.

Cuadro 4. Representación de la RdPCT Sistema (fig. 3) en forma de tuplas.

Colección de colores	$\square =$	{U, UT, TI, Tipo, Dato}																		
Lugares	$P =$	{Conteo, Productos, Cola in M1, M1 Libre, Maq M1, Cola out M1, Cola in M2, M2 Libre, Maq M2, Cola out M2, Cola in M3, M3 Libre, Maq M3, Cola out M3}																		
Transiciones	$T =$	{Llegadas, Inicio M1, Salida M1, Inicio M2, Salida M2, Inicio M3, Salida M3, Robot 1, Robot 2}																		
Arcos	$A =$	{(Conteo, Llegadas), (Llegadas, Conteo), (Productos, Llegada), (Llegadas, Producto), (Llegadas, Cola in M1), (Cola in M1, Inicio M1), (Inicio M1, Maq M1), (Maq M1, Salida M1), (Salida M1, Cola out M1), (Salida M1, M1 Libre), (M1 Libre, Inicio M1), (Cola in M2, Inicio M2), (Inicio M2, Maq M2), (Maq M2, Salida M2), (Salida M2, Cola out M2), (Salida M2, M2 Libre), (M2 Libre, Inicio M2), (Cola in M3, Inicio M3), (Inicio M3, Maq M3), (Maq M3, Salida M3), (Salida M3, Cola out M3), (Salida M3, M3 Libre), (M3 Libre, Inicio M3), (Cola out M1, Robot 1), (Robot 1, Cola in M2), (Cola out M2, Robot 2), (Robot 3, Cola in M3), }																		
Función de nodos	$N(a) =$	(Origen, Destino) Si a tienen la forma de par ordenado (x,y)																		
Función de color	$C(p) =$	<table border="0"> <tr> <td>Tipo</td> <td>Si p = Productos</td> </tr> <tr> <td>UT</td> <td>Si p = Conteo</td> </tr> <tr> <td>U</td> <td>Si p {M1 Libre, M2 Libre, M3 Libre}</td> </tr> <tr> <td>Dato</td> <td>Si p { Cola in M1, Maq M1, Cola out M1, Cola in M2, Maq M2, Cola out M2, Cola in M3, Maq M3, Cola out M3}</td> </tr> </table>	Tipo	Si p = Productos	UT	Si p = Conteo	U	Si p {M1 Libre, M2 Libre, M3 Libre}	Dato	Si p { Cola in M1, Maq M1, Cola out M1, Cola in M2, Maq M2, Cola out M2, Cola in M3, Maq M3, Cola out M3}										
Tipo	Si p = Productos																			
UT	Si p = Conteo																			
U	Si p {M1 Libre, M2 Libre, M3 Libre}																			
Dato	Si p { Cola in M1, Maq M1, Cola out M1, Cola in M2, Maq M2, Cola out M2, Cola in M3, Maq M3, Cola out M3}																			
Guardas	$G(t) =$	<table border="0"> <tr> <td>n=k1</td> <td>Si t {Inicio M1, Inicio M2, Inicio M2}</td> </tr> <tr> <td>TRUE</td> <td>Cualquier otro t</td> </tr> </table>	n=k1	Si t {Inicio M1, Inicio M2, Inicio M2}	TRUE	Cualquier otro t														
n=k1	Si t {Inicio M1, Inicio M2, Inicio M2}																			
TRUE	Cualquier otro t																			
Expresiones de los arcos	$E(a) =$	<table border="0"> <tr> <td>(k+1)@+tiempo(5,4)</td> <td>Si a = (Llegadas, Conteo)</td> </tr> <tr> <td>k</td> <td>Si a = (Conteo, Llegadas)</td> </tr> <tr> <td>p</td> <td>Si a {(Productos, Llegada), (Llegadas, Producto)}</td> </tr> <tr> <td>n</td> <td>Si a {(M1 Libre, Inicio M1), (M2 Libre, Inicio M2), (M3 Libre, Inicio M3)}</td> </tr> <tr> <td>k1+1</td> <td>Si a {(M1 Libre, Inicio M1), (M2 Libre, Inicio M2), (M3 Libre, Inicio M3)}</td> </tr> <tr> <td>(k1,p1,t1)</td> <td>Si a {(Cola in M1, Inicio M1), (Cola in M2, Inicio M2), (Cola in M3, Inicio M3), (Llegadas, Cola in M1)}</td> </tr> <tr> <td>(k1,p1,t1)</td> <td>Si a {(Inicio M1, Maq M1), (Maq M1, Salida M1), (Inicio M2, Maq M2), (Maq M2, Salida M2), (Inicio M3, Maq M3), (Maq Salida M3)}</td> </tr> <tr> <td>M3,</td> <td></td> </tr> <tr> <td>(k1,p1,t2)</td> <td>Si a {(Salida M1, Cola out M1), (Salida M2, Cola out M2), (Salida M3, Cola out M3),}</td> </tr> </table>	(k+1)@+tiempo(5,4)	Si a = (Llegadas, Conteo)	k	Si a = (Conteo, Llegadas)	p	Si a {(Productos, Llegada), (Llegadas, Producto)}	n	Si a {(M1 Libre, Inicio M1), (M2 Libre, Inicio M2), (M3 Libre, Inicio M3)}	k1+1	Si a {(M1 Libre, Inicio M1), (M2 Libre, Inicio M2), (M3 Libre, Inicio M3)}	(k1,p1,t1)	Si a {(Cola in M1, Inicio M1), (Cola in M2, Inicio M2), (Cola in M3, Inicio M3), (Llegadas, Cola in M1)}	(k1,p1,t1)	Si a {(Inicio M1, Maq M1), (Maq M1, Salida M1), (Inicio M2, Maq M2), (Maq M2, Salida M2), (Inicio M3, Maq M3), (Maq Salida M3)}	M3,		(k1,p1,t2)	Si a {(Salida M1, Cola out M1), (Salida M2, Cola out M2), (Salida M3, Cola out M3),}
(k+1)@+tiempo(5,4)	Si a = (Llegadas, Conteo)																			
k	Si a = (Conteo, Llegadas)																			
p	Si a {(Productos, Llegada), (Llegadas, Producto)}																			
n	Si a {(M1 Libre, Inicio M1), (M2 Libre, Inicio M2), (M3 Libre, Inicio M3)}																			
k1+1	Si a {(M1 Libre, Inicio M1), (M2 Libre, Inicio M2), (M3 Libre, Inicio M3)}																			
(k1,p1,t1)	Si a {(Cola in M1, Inicio M1), (Cola in M2, Inicio M2), (Cola in M3, Inicio M3), (Llegadas, Cola in M1)}																			
(k1,p1,t1)	Si a {(Inicio M1, Maq M1), (Maq M1, Salida M1), (Inicio M2, Maq M2), (Maq M2, Salida M2), (Inicio M3, Maq M3), (Maq Salida M3)}																			
M3,																				
(k1,p1,t2)	Si a {(Salida M1, Cola out M1), (Salida M2, Cola out M2), (Salida M3, Cola out M3),}																			

Continúa...

Continuación Cuadro 4.

Función de inicialización	$I(p)=$	$\begin{cases} 1 \text{ (1)} \\ 1 \text{ ("A")+1 ("B")+1 ("C")} \\ \emptyset \end{cases}$ <p>Si $p \in \{\text{Conteo, M1 Libre, M2 Libre, M3 Libre}\}$ Si $p = \text{Producto}$ Cualquier otro p</p>
Función de retardo de tiempo	$D(t)=$	$\begin{cases} @+(\text{case } p1 \text{ of} \\ \quad \text{"A"} \Rightarrow \text{tiempo}(20,1) \\ \quad \text{"B"} \Rightarrow \text{tiempo}(10,1) \\ \quad \text{"C"} \Rightarrow \text{tiempo}(15,1)) \\ @+(\text{case } p1 \text{ of} \\ \quad \text{"A"} \Rightarrow \text{tiempo}(10,1) \\ \quad \text{"B"} \Rightarrow \text{tiempo}(15,1) \\ \quad \text{"C"} \Rightarrow \text{tiempo}(20,1)) \\ @+(\text{case } p1 \text{ of} \\ \quad \text{"A"} \Rightarrow \text{tiempo}(5,1) \\ \quad \text{"B"} \Rightarrow \text{tiempo}(20,1) \\ \quad \text{"C"} \Rightarrow \text{tiempo}(10,1)) \\ 0 \end{cases}$ <p>Si $t = \text{Inicio M1}$ Si $t = \text{Inicio M2}$ Si $t = \text{Inicio M3}$ Cualquier otro t</p>

La RdPC Sistema (figura 3) se especificó en forma de tuplas, dicha representación matemática se muestra en el cuadro 4. En este cuadro se muestran todos los colores usados en la red, los nodos lugares y transiciones, arcos dirigidos, funciones de color, guardas, expresiones de los arcos y funciones de inicialización, todos estos conjuntos y funciones fueron definidos según la sección previa. Sin embargo, la herramienta CPNTool implementa dos tipos de funciones llamadas retardo de tiempo y segmento de código, por tanto, CPNTool amplía la definición de la RdPC establecida en [4].

La función de retardo de tiempo, es una función que se asocia a cada transición de la red. Cuando se dispara la transición, la función suma a la estampa de tiempo de la marca (*token*) el tiempo que determina la función; esta marca no puede ser removida del nodo lugar posterior hasta que el tiempo simulado general del sistema es mayor o igual al nuevo valor calculado

de la estampa de tiempo. Esto implica que estas funciones determinan cuánto tiempo debe permanecer una marca en un nodo lugar antes de ser removida. Las funciones de retardo se reconocen porque tienen la siguiente estructura $@+(\text{código CPN ML})$. En la figura 3 y en el cuadro 4 se observa que las funciones de retardo llaman a la función $\text{tiempo}(\mu, \nu)$. Esta función implementa la duración de los tiempos según la distribución normal, tal y como se planteó en el problema. Los argumentos de entrada para la función son la media y la varianza.

La herramienta CPNTool crea las funciones llamadas "segmento de código". Estas funciones son útiles para calcular nuevos valores de datos, capturar tiempos del sistema y, especialmente, comunicarse con otros programas. Esta función se codifica con *CPN ML* y con funciones propias de BRITNeY. De manera que, para generar de forma automática el diagrama de gantt, el diagrama de colas, etc., CPNTool debe

El modelo del sistema de manufactura se simuló varias veces con la idea de verificar el comportamiento dinámico y la estructura de la red. A continuación se presentan resultados de una simulación rápida de tan solo 91 pasos ejecutados y un tiempo simulado de 86 unidades de tiempo.

transferir los datos al software BRITNeY mediante el segmento de código, si desea conocer mayor detalle puede consultar [13]. Cada función segmento de código está ligada a un determinado nodo transición, por ejemplo, en la figura 4, el segmento código de la transición “Sensor cola Punto 1”, aparece en la esquina superior izquierda de la figura. Esta función posee tres partes principales: *input()*, o lista de argumentos en la que cada argumento debe ser una variable existente en las expresiones de los arcos que entrada al nodo; *output()* o lista de argumentos de salida, que son variables que aparecen o sustituyen a otras variables en las expresiones de los arcos de salida del nodo; y *action* que es la sección del segmento de código donde se declaran las funciones y operaciones de cálculo, la cual se llama función *gantt.addTaskMinute()* pues grafica el gantt y captura el instante de tiempo simulado en que ocurrió la transición. Una explicación a profundidad sobre el uso del segmento de código esta disponible en [10].

Simulación

El modelo del sistema de manufactura se simuló varias veces con la idea de verificar el comportamiento dinámico y la estructura de la red. A continuación se presentan resultados de una simulación rápida de tan solo 91 pasos ejecutados y un tiempo simulado de 86 unidades de tiempo. Simulaciones con una mayor cantidad de pasos generan un volumen de información tal que el diagrama de gantt no es legible.

La figura 5 muestra un diagrama gantt generado automáticamente, el eje equis representa el tiempo simulado. Así, una simulación de 80 unidades de tiempo, se representa en el diagrama como 01:20. En el eje ya aparecen actividades del proceso productivo. Cada actividad listada está asociada a un nodo lugar de la RdPC, esto significa que la actividad Proceso M1 del gráfico captura el comportamiento de las

piezas que pasan por el nodo lugar Maq M1, a lo largo del tiempo de simulación. Es importante aclarar que la forma en que se programó la generación del diagrama gantt, permite graficar únicamente piezas que han finalizado una actividad determinada, por tanto, una actividad que se esté ejecutando no aparece en el diagrama.

Las piezas que ingresan al sistema se identifican mediante un número ascendente seguido del tipo de producto. Por ejemplo, el diagrama gantt se construyó con las siguiente siete piezas: 1B, 2A, 3B, 4B, 5C, 6A, 7C. Cada pieza se representa en el diagrama como barras de un color: la pieza 1B se muestra como el conjunto de barras color rojo, 2A son las barras color verde claro, y así sucesivamente. En el diagrama gantt aparecen otras barras nombradas como 1T, 2T, 3T, 4T, 5T. Las barras indican que los robots no están transportando piezas debido a que se encuentran en uno de los siguiente dos estados: o en reposo, o que están retornando al punto de inicio, (Punto 1).

La figura 5 muestra el recorrido de la pieza 1B por todo el sistema mediante flechas color negro; al estar el sistema en un estado inicial no existen colas de espera y todas las máquinas y robots están disponibles, por tanto se puede observar que esta pieza nunca estuvo en colas. Por otro lado, cuando la pieza 2A ingresa al sistema, entra directamente a la cola de la máquina M1, puesto que la máquina M1 se encuentra procesando la pieza 1B. La pieza 2A es procesada hasta que M1 se desocupe. El recorrido de la pieza 2A por todo el sistema, se señala con flechas rayadas. Se representa en la figura trazos de líneas punteadas y rayas que muestran la operación de ambos robots, principalmente desocupación y ocupación.

Si se toma la actividad Cola M1 y se grafica sólo el comportamiento de esa actividad, se obtiene la gráfica 6, la cual se conoce como gráfica de evolución de cola. Esta gráfica es una muestra, ya

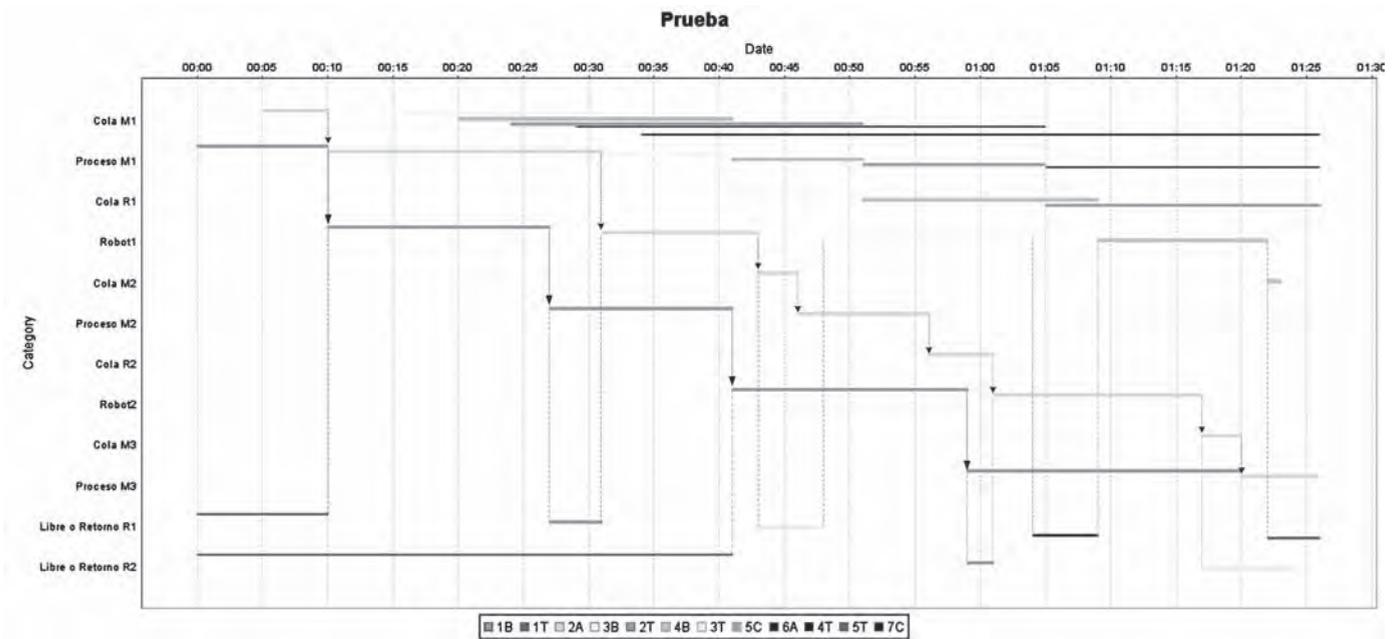


Figura 5. Diagrama de Gantt obtenido por simulación. Los productos son tres tipos: A, B, C y están numerados ascendentemente. En la gráfica aparece 1T, 2T, 3T, etc., estas barras corresponden a lapsos de tiempo, donde los robots están sin trabajo o en el proceso de retorno al punto uno.

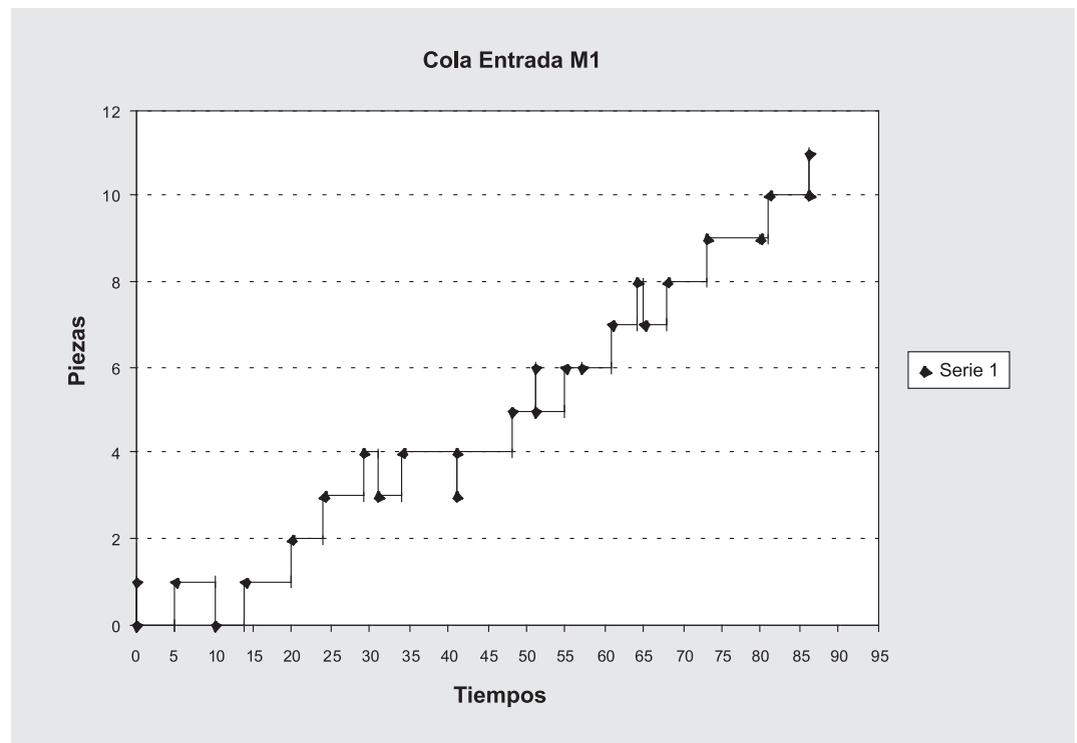


Figura 6. Diagrama que muestra el comportamiento de la cola de entrada a la máquina M1. El área bajo la curva brinda el promedio de piezas en la cola. Este gráfico fue realizado en Microsoft Excel, los datos fueron exportados de CPNTool en un archivo de texto.

que se pueden obtener las seis gráficas de cola del sistema al graficar los datos capturados por los monitores (funciones de CPNTool). Esto se logra al instalar una función monitor en cada nodo lugar en que se desee conocer los instantes de tiempo en que cambia el conteo del nodo. El historial luego es exportado en forma de un archivo de texto que posteriormente se analiza con MS Excel. El área bajo la curva de la gráfica representa la cantidad de piezas en espera promedio, para esta cola es de 4,53 piezas.

Cuadro 5. Estadísticas generadas en la simulación del sistema de manufactura.

Nodo Lugar	Indice	Explicación
Cola in M1	4,534884	Promedio de piezas en la cola de entrada de M1.
Cola in M2	0,081395	Promedio de piezas en la cola de entrada de M2.
Cola in M3	0,034884	Promedio de piezas en la cola de entrada de M3.
M1 Libre	0,000000	Máquina M1 no tiene desocupación.
M1 Ocupada	1,000000	Máquina M1 esta en 100% de utilización.
M2 Libre	0,523256	Máquina M2 tiene un 52,3% del tiempo en desocupación.
M2 Ocupada	0,476744	Máquina M2 tiene un 47,7% del tiempo procesando.
M3 Libre	0,686047	Máquina M3 tiene un 68,6% del tiempo en desocupación.
M3 Ocupada	0,313953	Máquina M2 tiene un 31,4% del tiempo procesando.
R1 Libre	0,325581	El 32,6% del tiempo R1 esta desocupada o devolviéndose al punto 1.
R1 Ocupado	0,674419	El 67,4% del tiempo R1 traslada piezas del punto 1 al punto 6.
R2 Libre	0,581395	El 58,1% del tiempo R1 esta desocupada o devolviéndose al punto 1.
R2 Ocupado	0,418605	El 41,9% del tiempo R2 traslada piezas del punto 1 al punto 6.

La herramienta CPNTool permite calcular una serie de estadísticas, promedios ponderados, desviaciones estándar, porcentajes de utilización, tiempos promedios de espera en cola, etc.; sin embargo, muchas de estas estadísticas están disponibles si los datos dentro de los nodos lugares son declarados en listas. El manejo de datos que se utilizó en el sistema está basado en el concepto de tupla, por lo que si se desea calcular, por ejemplo, el tiempo promedio de espera de una pieza, hay que programar la función que la implemente. El cuadro 5 presenta estadísticas del sistema, las cuales fueron definidas en los objetivos del proyecto. De la figura 5 se pueden hacer conclusiones sobre los porcentajes de utilización; por ejemplo, según los cálculos de la herramienta, la máquina M1 trabaja a un 100%, sin embargo no da abasto, puesto que los tiempos de duración programados para esta máquina son superiores a la tasa de arribo de nuevas piezas, lo que causa un cuello de botella.

Recomendaciones

Dado que los softwares utilizados son de licenciamiento libre, con potentes capacidades de edición, simulación y reporte, es recomendable establecer proyectos de investigación en las siguientes líneas:

- Desarrollar metodologías que permitan un adecuado proceso de modelado de sistemas de eventos discretos con RdPC y otras familias.
- Extender el desarrollo científico tecnológico al medio industrial, de tal forma que la utilización de modelos simulados con RdPC permita mejorar los sistemas de manufactura reales, y elimine desbalances de producción, bloqueos, subutilización de sistemas, etc.
- Establecer procedimientos de verificación y validación para los modelos de simulación.

En cuanto al modelo planteado, éste se podría mejorar si se realizan los siguientes cambios:

- Modificar la topología de la red, de tal forma que cada máquina y cada robot se puedan trabajar como marcas (objetos) dentro de la red, lo cual permitirá un modelo más compacto. Sin embargo, esto dificultaría la compresión del modelo.
- Establecer un mayor nivel de abstracción, al utilizar listas en lugar de tuplas para el manejo de los objetos en los nodos lugar, esto permitiría al CPNTool calcular el retardo promedio de un objeto en la cola, etc.
- Establecer límites a las colas de entrada a cada máquina y a cada robot, con la finalidad de observar el bloqueo del sistema.

Bibliografía

- [1]. Bell Labs, Lucent Technologies, *The Standard ML Basis Library*, Sitio: <http://www.dcs.napier.ac.uk/~andrew/SMLBasis/sml-std-basis.html>
- [2]. Guasch, A. *et al.*, (2005). *Modelado y simulación. Aplicación a procesos logísticos de fabricación y servicios*. Mexico, Alfaomega.
- [3]. Hruz, B., *et al.*, (2007). *Modeling and Control of Discrete-event Dynamic Systems*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag, London.
- [4]. Jensen, K., (1992). Coloured Petri Nets. Basic Concept. Analysis Methods and Practical Use, Vol 1, Vol 2, *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag. Berlin.
- [5]. Jensen, K., (1994). An Introduction to the Theoretical Aspects of Coloured Petri Nets. *A Decade of Concurrency, Lecture Notes in Computer Science* vol. 803, Springer-Verlag, 230-272. Fuente: http://www.daimi.au.dk/~kjensen/papers_books/rex.pdf
- [6]. Jensen, K., (1998). An Introduction to the Practical Use of Coloured Petri Nets, *LNCS: Lecture on Petri Nets Application II*, vol 1492, Berlin: Springer-Verlag, 237-292. Fuente: http://www.daimi.au.dk/~kjensen/papers_books/use.pdf
- [7]. Jensen, K., *et al.*, (Junio 2007). Coloured Petri Nets and CPN Tools for modeling and validation of concurrent systems, *International Journal on Software Tools for Technology Transfer (STTT)*, vol 9, No 3-4, Berlin: Springer-Verlag, 213-254. Fuente: http://www.daimi.au.dk/~kjensen/papers_books/sttt2007.pdf
- [8]. Silva, M., *et al.*, (1998). Petri nets and Production Systems, *LNCS: Lecture on Petri Nets Application II*, vol 1492, Berlin: Springer-Verlag, 85-124.
- [9]. Stöle, H., (1998). An Evaluation of High-End Tools for Petri Nets, Technical Report 9802, University of Munich, 1998. Sitio: <http://www.pst.informatik.uni-muenchen.de/personen/stoerle/V/Evaluierung.pdf>
- [10]. University of Aarhus. Denmark, *Computer Tool for Coloured Petri Nets (CPNTOOL)*, Fuente: http://wiki.daimi.au.dk/cpntools/_home.wiki
- [11]. University of Hamburg. Germany, *et al*, *Petri Nets World*, Fuente: <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- [12]. Vinter, A., *et al.*, (2003). CPN Tools for Editing, Simulating and Analysing Coloured Petri Nets, *Lecture Notes in Computer Science: ICATPN 2003*, vol 2679, Berlin: Springer-Verlag, 450-462. Fuente: <http://citeseerx.ist.psu.edu/viewdoc/summary.jsessionid=8278F4ECFF9373D19D45C7C21EBFBC2C?doi=10.1.1.87.7860>
- [13]. Westergaard, M., *et al.*, (Octubre 2007). Building and Deploying Visualizations of Coloured Petri Net Models Using BRITNeY Animation and CPN Tools, *Proceedings of Sixth Workshop on Practical Use of Coloured Petri Nets and the CPN Tools*. Århus, Denmark. Fuente: <http://klafbang.eu/personlig/publications.php3>
- [14]. Westergaard, M. *BRITNeY Suite*, Fuente: <http://wiki.daimi.au.dk/britney/britney.wiki>