

ESTRUCTURAS GENERALIZADAS PARA CONTROLADORES LÓGICOS MODELADAS MEDIANTE REDES DE PETRI

GERMAN ZAPATA

Escuela de Ingeniería Eléctrica y Mecánica. Universidad Nacional de Colombia sede Medellín.

ELIANA CARRASCO

Instituto Tecnológico Metropolitano.

Recibido para revisar 17 de Abril de 2002, aceptado Junio 28 de 2002, versión final 24 de Septiembre de 2002.

RESUMEN: Se presenta una metodología para el diseño estructurado de automatismos secuenciales, empleando como herramienta de modelamiento y simulación las Redes de Petri. Una vez planteados los modelos de estructuras generalizadas, así como su análisis y validación, se presentan algunos aspectos relacionados con el mapeo de estas estructuras a lenguajes normalizados de controladores programables, según IEC 61131. Se abordan también consideraciones respecto a la implementación en un controlador específico

PALABRAS CLAVES: Automatismos Secuenciales, Sistemas Secuenciales, Redes de Petri, Controladores Lógicos Programables.

ABSTRACT: This work presents a methodology for the structural design of sequentials automaton, using Petri Nets as a modeling and simulation tool. After introducing the models for generalized structures and their analysis and validation, some aspects related to the mapping of such structures to normalized languages are presented in accordance to IEC 61131. Finally, considerations about structure implementation on specific are exposed.

KEYWORDS: Sequentials Automaton, Sequentials Systems, Petri's Nets, Logic Programming Controllers.

1 INTRODUCCIÓN

En el desarrollo de proyectos de automatización industrial, el equipo de control más utilizado ha sido indudablemente el controlador lógico programable. Los avances en la industria del software han permitido que la aplicación de metodologías de diseño estructurado y los paradigmas de la programación orientada a objetos puedan ser llevadas a los lenguajes de programación de estos controladores. Este trabajo presenta unos modelos generales de los dispositivos presentes en muchos

de los procesos industriales, planteados como estructuras de programación, que pueden ser llevados a bloques funcionales parametrizables, obteniendo con ello grandes beneficios en la estructuración de las tareas de los automatismos. Para el diseño de las estructuras se empleó una herramienta de modelación de sistemas a eventos discretos, denominada Redes de Petri (RdP). Después de la presentación teórica de esta herramienta se ilustra su aplicación en el diseño de tres estructuras, las cuales son traducidas a un lenguaje de programación bajo la norma

IEC61131. El paso del modelo al lenguaje es explicado brevemente. Finalmente, con el fin de ilustrar la aplicación de la metodología, se presenta el programa del automatismo de un proceso industrial simplificado. El método propuesto, basado en RdP, tiene ventajas sobre otros métodos convencionales, como la programación en lógica de relés, sobre todo en los aspectos de permitir el diseño sistemático y estructurado, la flexibilidad, la posibilidad de efectuar análisis del desempeño y validación formal del diseño.

2 LAS REDES DE PETRI Y EL CONTROL SECUENCIAL

El avance y crecimiento de las Redes de Petri (RdP) ha sido notable desde su desarrollo en 1962. Cuenta de ello, dan las numerosas publicaciones que muestran trabajos en el modelamiento de sistemas concurrentes, protocolos de comunicaciones, sistemas de tiempo real, sistemas distribuidos, sistemas de manufactura, sistemas multiprocesador y por supuesto, sistemas secuenciales. Referencias muy completas de aplicaciones y producción académica se pueden encontrar en (Murata, 1998), (Zarawski *et al.*, 1994), (Silva, 1982) y (Paterson, 1981).

Posterior a la aparición de los PLC's (Programmable Logic Controller), para la realización de tareas secuenciales en un proceso industrial y del primer lenguaje de programación basado en lógica de relés, los métodos basados en RdP fueron ganando auge: el GRAFCET (Davis, 1992) (después denominado SFC Sequential Function Chart por IEC), tiene como fundamento las RdP y otros métodos como los diagramas estado-transición, cartas de estado, máquinas de estados finitos, son casos particulares de RdP. (Silva, 1982)

Existen muchos trabajos que presentan las RdP como herramientas para modelar, analizar, evaluar desempeños y controlar sistemas a eventos discretos, dentro de los cuales cabe mencionar (Desrochers, 1995) y (Zhou, 1996). Se destaca allí que las RdP tienen gran potencial para modelar sistemas secuenciales, asíncronos, de eventos concurrentes muy comunes en los procesos industriales, de ahí que sean utilizados para

controlar dichos procesos. En (Zhou, 1998), se presenta una bibliografía muy completa sobre métodos de diseño y demostraciones de laboratorio, así como herramientas computacionales para aplicación industrial que facilitan el diseño asistido (CAD). Ese trabajo muestra una estrategia de control que usa un modelo en RdP que captura las dinámicas de eventos discretos del proceso controlado, lo que resulta en un diseño fácil de entender, reparar, modificar y evaluar.

La preocupación constante de los investigadores en el área de aplicación de las RdP en el control secuencial ha sido la de plantear metodologías que sistematicen y faciliten el diseño y la implantación, sin sacrificar robustez. Algunas ideas al respecto se encuentran en (Silva, 1982), que dedica un capítulo a la implementación con PLC's; en (Zhou, 1994) se presenta un método para traducir modelos en RdP a la lógica de contactos y el mismo autor (Zhou, 1995) hace la presentación de un método para generar automáticamente el código de programas secuenciales. Otros enfoques pueden ser consultados en (Frey, 1995), (Lennartson, 2000) y (Jiang, 1996), en los que se hace énfasis en la validación y en la especificación y diseño de los controladores en la parte discreta de sistemas híbridos.

La complejidad y crecimiento de los sistemas industriales automatizados, así como el hardware disponible (16 / 32 bits), ha permitido la utilización de RdP de alto nivel para la programación estructurada y orientada a objetos. Una extensión de estas redes, son las llamadas coloreadas (CPN - Coloured Petri Nets), que se presentan en (Kurt, 1997). Aplicaciones de las CPN en el control lógico se detallan en (Ezpeleta, 1997) y (Feldmann, 1999).

Se encuentran también métodos para traducir programas en GRAFCET o SFC a lenguaje de contactos; estas metodologías son muy apropiadas debido a que el lenguaje SFC se basa en las RdP y por lo tanto, resultan muy ilustrativas para su comprensión. Cabe resaltar las siguientes referencias: (Moreno), (Ferreiro, 1995), (Adamski, 1998), (Miyazawa, 1997). Un caso concreto de aplicación al PLC S7-200 de Siemens, se

presenta en (Pérez, 2000), por parte del grupo GENIA de la Universidad de Oviedo. En el artículo se hace énfasis en los problemas de implementación al pasar un modelo en SFC a PLC. Se destaca en esta revisión, el interés de la comunidad académica de buscar y plantear metodologías de diseño y especificación de sistemas secuenciales que sean aceptadas por la aplicación industrial.

Una tendencia parece ser la programación de PLC's mediante métodos basados en RdP (PN Based Logic Controller) (Frey, 1995) y (Pollard, 1995).

3 RDP TEMPORIZADAS E INTERPRETADAS

Con el fin de hacerlas más apropiadas para modelar sistemas secuenciales, controlados mediante PLC, las RdP definidas en (Murata, 1998), (Silva, 1982), (Paterson, 1981) han sido aumentadas, asignando atributos de tiempo y de procesamiento lógico a las transiciones y funciones de entrada/salida asociadas a los lugares para establecer las relaciones de la periferia del proceso (sensores y actuadores) con el modelo.

Definición :

Una Red de Petri temporizada e interpretada es una tupla

$$\mathbf{R} = (\mathbf{P}, \mathbf{T}, \mathbf{I}, \mathbf{Q}, \mathbf{D}, \mathbf{E}, \mathbf{C}, \mathbf{M}),$$

donde:

P es un conjunto finito no vacío de lugares.
T es un conjunto finito no vacío de transiciones.

I: $\mathbf{P} \times \mathbf{T} \rightarrow \mathbf{N}$ es una función de entrada que define un conjunto de arcos dirigidos de **P** a **T** de peso **N**.

Q: $\mathbf{P} \times \mathbf{T} \rightarrow \mathbf{N}$ es una función de salida que define un conjunto de arcos dirigidos de **T** a **P** de peso **N**.

D: $\mathbf{T} \rightarrow \mathbf{R}^+$ es una función de tiempo, que asigna a cada transición un retardo.

E: $\mathbf{T} \rightarrow \Sigma$ es una función que mapea a cada transición un evento perteneciente a un alfabeto Σ .

El evento es el cambio del estado lógico de una variable del tipo binario, sea de entrada, de salida o estado interno. Al evento positivo (cambio de 0 a 1) de la variable **a** se designará $\alpha = \uparrow a$. Al evento negativo de **a** se designará $\beta = \downarrow a$. El alfabeto $\Sigma = \{\alpha, \beta, \delta, \dots\}$.

C: $\mathbf{T} \rightarrow \mathbf{Bexpr}$ es una función que asigna a cada transición una expresión booleana o predicado. El valor de **Bexpr** se denota **Val(Bexpr)**. $\text{Val}(Bexpr) \in \{0,1\}$. El producto $E(t_i) \bullet C(t_i)$ se denomina receptividad de la transición *i*.

M: $\mathbf{P} \rightarrow \mathbf{N}$ Marcaje donde la *i*-ésima componente representa el número de marcas del *i*-ésimo lugar. Un marcaje inicial se denota por M_0 .

La dinámica de la red está determinada por el flujo de marcas. El marcaje cambia en la red de acuerdo a las siguientes reglas de disparo:

- 1) Una transición *t* se dice que está habilitada o sensibilizada para el disparo si cada lugar **p** de entrada de *t* está marcado con al menos $I(p,t)$ marcas. Esto es:

$$t \in T \text{ está habilitada si } M(p) \geq I(p,t), \forall p \in P$$

- 2) Una transición *i* sensibilizada, dispara después de transcurrido el tiempo $D(t_i)$ si la receptividad asociada a ella se verifica. Esto es, si $E(t_i) \bullet C(t_i) \neq 0$.

- 3) El disparo de una transición sensibilizada consiste en remover $I(p,t)$ marcas de cada lugar de entrada y agregar $Q(p,t)$ marcas a cada lugar de salida.

Si $M_i \xrightarrow{t} M_j$ representa la evolución del marcaje desde M_i hasta M_j por el disparo de *t*, se tiene entonces la siguiente regla de evolución del marcaje:

$$M_j(p) = M_i(p) + Q(p, t) - I(p, t) \quad \forall p \in P$$

El fundamento de la modelación de los sistemas secuenciales mediante RdP es

concebir éstos como sistemas a eventos discretos, en los cuales el sistema evoluciona de un estado a otro por la ocurrencia de un evento.

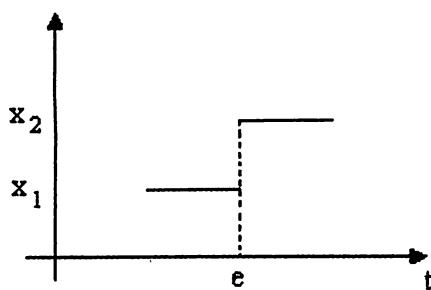


Figura 1. Sistema a eventos discretos.

En la Figura 1, el sistema evoluciona del estado x_1 al estado x_2 por la ocurrencia del evento e_1 .

El modelo que representa esta situación es el siguiente:

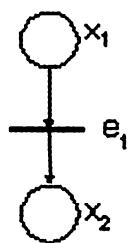


Figura 2. Esquema en RdP de la Figura 1.

Una RdP modela un SED (Sistema a Eventos Discretos), asociando los estados a los lugares y los eventos a transiciones. Por ejemplo, el modelo de la Figura 3 representa la función de memoria set/reset aplicada al arranque y paro de un motor:

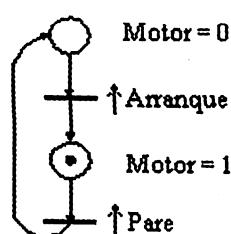


Figura 3. Función de arranque y paro de un motor.

El marcate indicará el estado actual del sistema. En la figura anterior, el marcate indica que el motor está encendido.

Una vez presentado el marco conceptual, se procede a presentar el diseño de las estructuras, en las que se empleó como fundamento, la teoría expuesta de las RdP.

4 ESTRUCTURAS GENERALIZADAS

La metodología que se propone para la solución de sistemas secuenciales utilizados en la automatización de procesos industriales identifica elementos comunes en los procesos, denominados dispositivos. Estos dispositivos interrelacionados conforman subprocessos, los que a su vez hacen parte de las áreas funcionales de los procesos, como se muestra en la Figura 4.

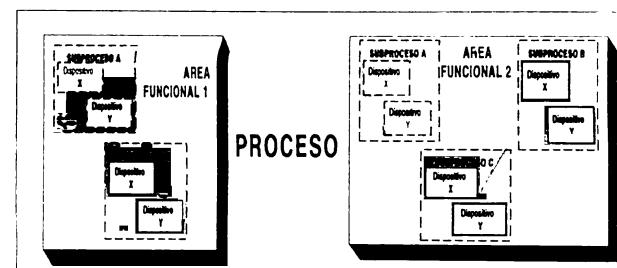


Figura 4. Estructura general de un proceso

Los dispositivos o estructuras generalizadas se modelan mediante RdP, se simulan y se validan mediante una herramienta formal de análisis, como los grafos de ocurrencia (*Murata*, 1989) y (*Kurt*, 1997). Posteriormente son codificadas en lenguajes de controladores lógicos como bloques funcionales parametrizables.

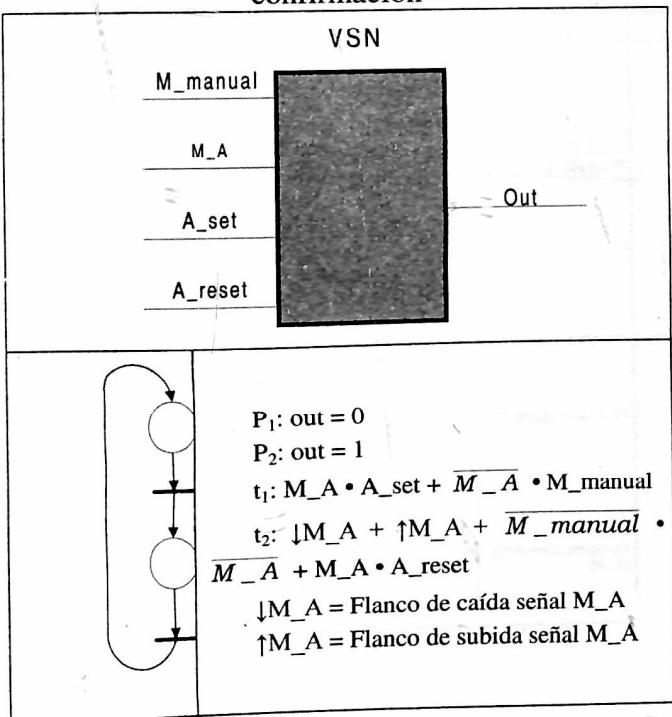
Los dispositivos son los diferentes actuadores utilizados en los procesos industriales, los cuales en su forma más básica poseen una señal de mando y una señal de salida que maneja un elemento final de control, existiendo gran cantidad de variantes dependiendo del tipo y la forma como operan. Los más utilizados se enumeran en la Tabla 1.

Tabla 1. Dispositivos

DISPO-SITIVO	SIM	ACCION
Válvulas	VSN	Simple efecto/sin confirmación
	VSS	Simple efecto/simple confirmación
	VSD	Simple efecto/doble confirmación
	VDN	Doble efecto/sin confirmación
	VDD	Doble efecto/doble confirmación
Motores	MSN	Simple mando/sin realimentación
	MSS	Simple mando/con realimentación
	MSR	Reversible forward/reverse
Cilindros	CSD	Actuador sencillo/doble realimentación. Existen dos variantes -Estado energizado extendido -Estado energizado retraído

La importancia del diseño estructurado permite considerar los mandos o modos de marcha, las seguridades y el control supervisorio. Los modelos y la representación como estructura generalizada de algunos de estos dispositivos se muestran en las Tablas 2 y 3.

Tabla 2. Dispositivo Simple efecto / sin confirmación



M_manual: Señal de activación en modo manual

M_A: Modo manual / automático

A_set: Señal activación en automático del dispositivo

A_reset: Señal desactivación en automático del dispositivo

Out: Salida dispositivo

Tabla 3. Dispositivo Doble efecto / sin confirmación

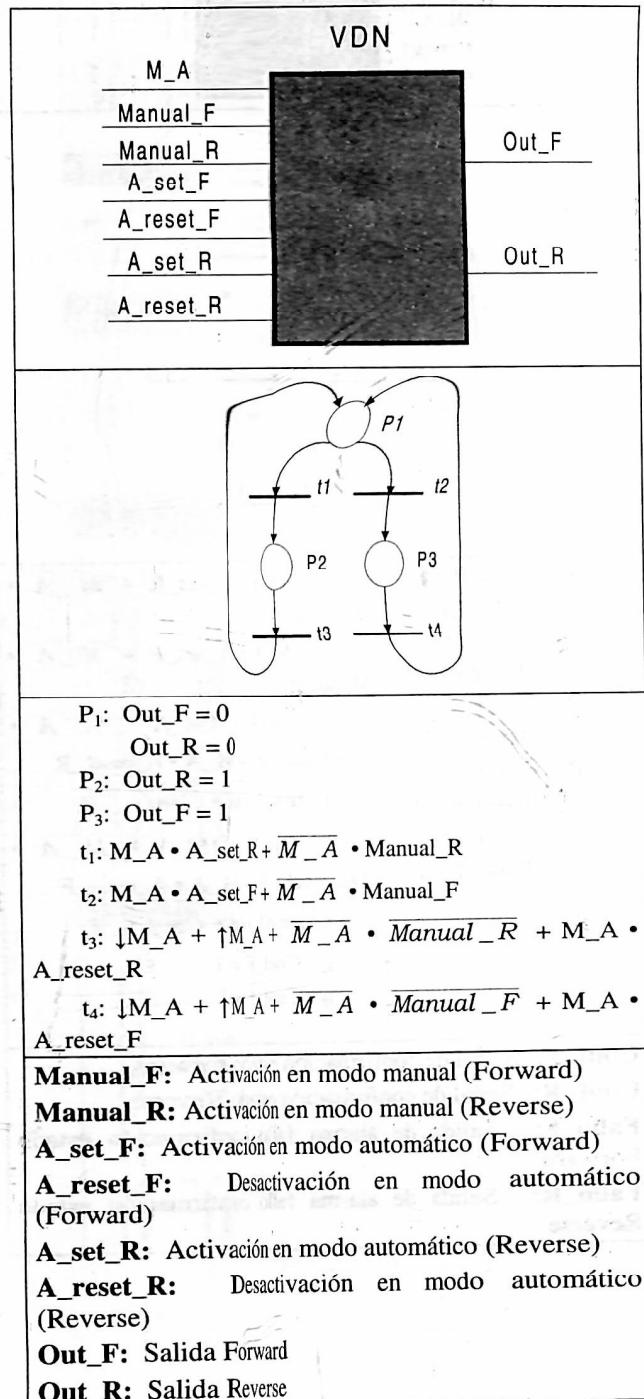


Tabla 4. Dispositivo Doble efecto/doble confirmación

P ₁ : Out_F = 0 Fallo_F = 0	t ₁ : M_A • A_set_R + $\overline{M_A}$ • Manual_R
P ₂ : Out_R = 1 Temp1 = ON	t ₂ : M_A • A_set_F + $\overline{M_A}$ • Manual_F
P ₃ : Out_F = 1 Temp2 = ON	t ₃ : $\downarrow M_A$ + $\uparrow M_A$ + $\overline{M_A}$ • Manual_R + M_A • A_reset_R
P ₄ : Out_R = 0 Fallo_R = 1	t ₄ : temp1_out • Conf_R
P ₅ : Out_F = 0 Fallo_F = 1	t ₅ : $\downarrow M_A$ + $\uparrow M_A$ + $\overline{M_A}$ • Manual_F + M_A • A_reset_F
	t ₆ : temp2_out • Conf_F
	t ₇ : Conf_R = 1
	t ₈ : Conf_F = 1
Conf_F: Señal de confirmación estado forward	
Conf_R: Señal de confirmación estado Reverse	
Fallo_F: Salida de alarma fallo confirmación estado Forward	
Fallo_R: Salida de alarma fallo confirmación estado Reverse	

5 MAPEO DE ESTRUCTURAS A LENGUAJES NORMALIZADOS (IEC 61131)

Los procedimientos de diseño de sistemas secuenciales mediante RdP deben contar además con una metodología para su implementación en equipos digitales. A nivel industrial, el equipo de uso más extensivo es el PLC (Programmable Logic Controller), por sus características de robustez, flexibilidad y su concepción como dispositivo de propósito general (Zhou, 1998). Para la traducción del modelo del sistema a un lenguaje de PLC normalizado se tendrán en cuenta las representaciones de la Tabla 5:

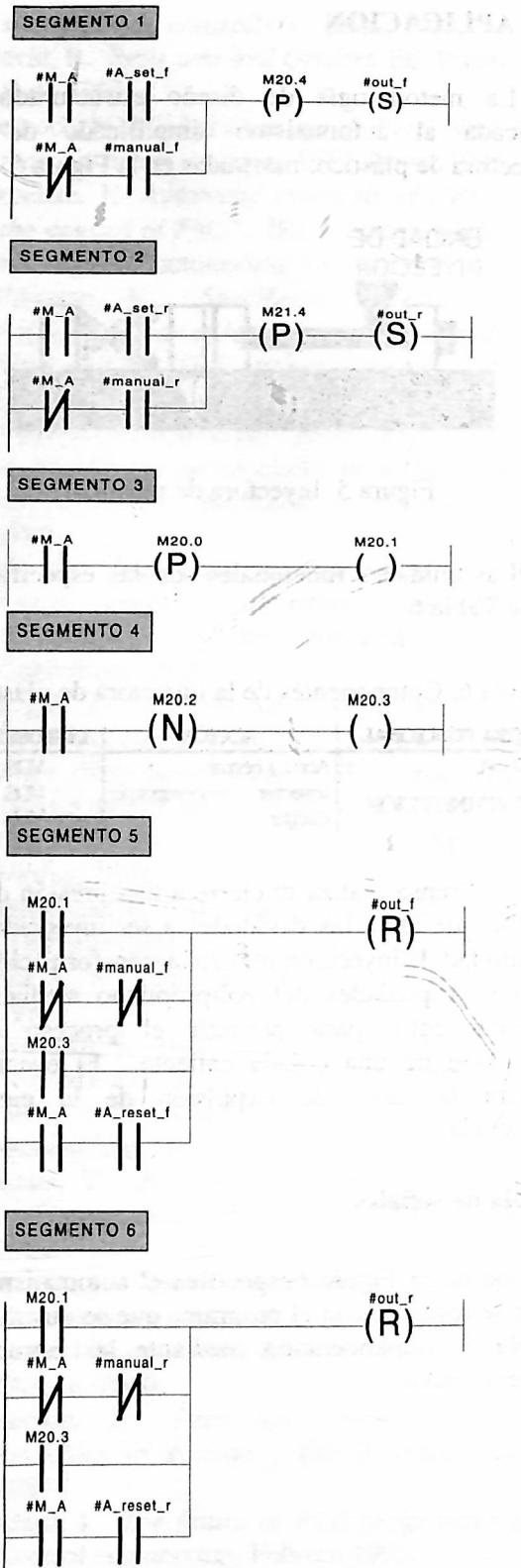
Tabla 5. Representación de las redes de Petri a RLL

CONSTRUCCIÓN LÓGICA	RdP	RLL (Relay Ladder Logic)
Condición o estado de un elemento del sistema		Sin representación
Actividad, acción o evento		Sin representación
Elemento activo		Sin representación
Función lógica "y"		A B C D --- --- --- --- ---
Función lógica "O"		A B C D
Función memoria (S/R)		S_A R_B

CONSTRUCCION LÓGICA	RdP	RLL (Relay Ladder)
Flanco		
Receptividad		
Concurrencia		
Retardo		

Por ejemplo, la traducción del programa de PLC para el dispositivo VDN se ve a continuación. Se utilizó para la codificación el STEP7 de SIEMENS suministrado para los equipos de la línea SIMATIC S7-300.

NOMBRE	COMENTARIOS
A_set	Señal de arranque automático
A_set	Señal de arranque automático
manual	Activación en modo manual
A_reset	Desactivación en modo automático
A_reset	Desactivación en modo automático
manual	Activación en modo manual
M_out	Selector
Out_	Salida
Out_	Salida
m1	



6 APLICACION

La metodología de diseño estructurado fue aplicada al automatismo simplificado de una inyectora de plástico, mostradas en la Figura 5.

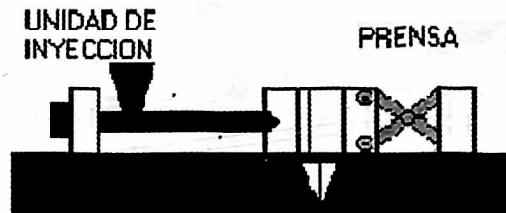


Figura 5. Inyectora de plástico.

Las unidades funcionales son las especificadas en la Tabla 6.

Tabla 6. Componentes de la inyectora de plástico.

ÁREA FUNCIONAL	ACIÓN	DISPOSITIVO
PRENSA	Abrir / cerrar	VDS
UNIDAD INYECCIÓN	Inyectar / descomprimir	VDS
	Cargar	VDN

La Prensa realiza el cierre a alta presión del molde que aloja las cavidades a ser inyectadas. La unidad de inyección realiza la transformación, de las propiedades del polipropileno mediante calentamiento, para permitir el proceso de inyección de una colada caliente. El eyector realiza la tarea de expulsión de la pieza inyectada.

Tabla de señales

La red de la Figura 6 especifica el automatismo, el cual se resuelve con el programa que se ilustra en la Tabla 7, implementado mediante las estructuras generalizadas.

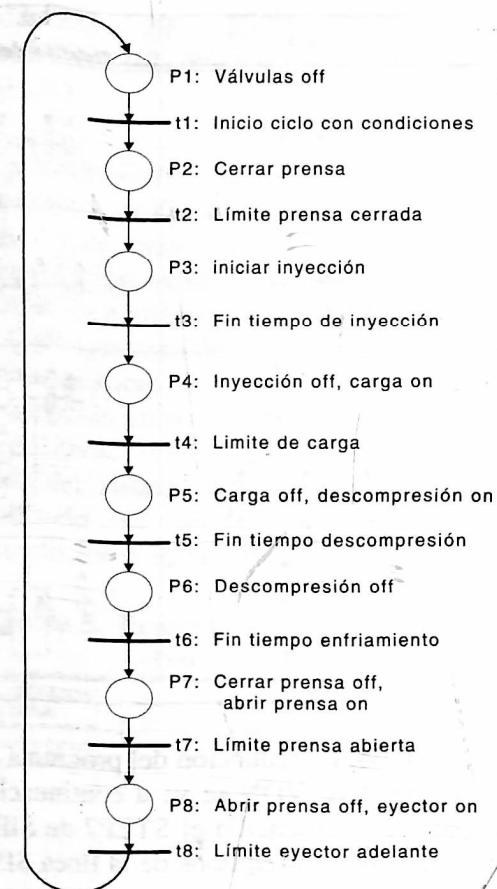
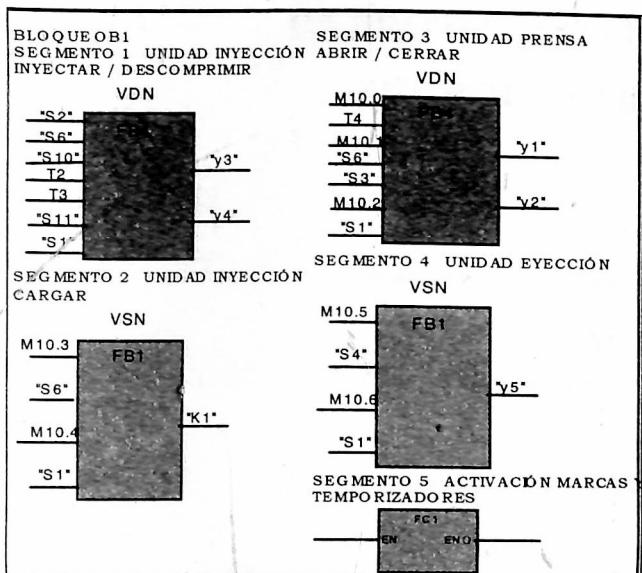


Figura 6. Ciclo automático inyectora de plástico

Tabla 7. Inventario de señales

SÍMBOLO	DIRECCIÓN	COMENTARIO
Y1	A 124.0	Solenoide cerrar prensa
Y2	A 124.1	Solenoide abrir prensa
Y3	A 124.2	Solenoide inyectar
Y4	A 124.3	Solenoide descomprimir
Y5	A 124.4	Solenoide eyector adelante/atrás
K1	A 124.5	Contactor motor carga
S0	E 124.0	Pulsador inicio de ciclo
S1	E 124.1	Selector manual/automático
S2	E 124.2	Microsuiche límite de carrea prensa cerrada
S3	E 124.3	Microsuiche límite de carrea prensa abierta
S4	E 124.4	Microsuiche límite de carrea eyector adelante
S5	E 124.5	Microsuiche límite de carrea eyector atrás
S6	E 124.6	Microsuiche límite de carga
S7	E 124.7	Microsuiche puerta cerrada
S8	E 125.0	Pulsador mando manual cerrar prensa
S9	E 125.1	Pulsador mando manual abrir prensa
S10	E 125.2	Pulsador mando manual inyectar
S11	E 125.3	Pulsador mando manual descomprimir
S12	E 125.4	Pulsador mando manual cargar
S13	E 125.5	Pulsador mando manual eyector
T 1		Tiempo de pausa
T 2		Tiempo de inyección
T 3		Tiempo de descompresión
T 4		Tiempo de enfriamiento
M10.3		Marca habilitar cargar en automático
M10.4		Marca habilitar cargar modo manual hasta límite de recorrido
M10.0		Marca habilitar cerrar prensa en automático
M10.1		Marca habilitar cerrar modo manual hasta límite de recorrido
M10.2		Marca habilitar abrir modo manual hasta límite de recorrido
M10.5		Marca habilitar eyector en automático
M10.6		Marca habilitar eyector modo manual hasta límite de recorrido

Programa de la Inyectora



7 CONCLUSIONES

El poder de representación de las RdP y su robustez matemática han aportado los procedimientos de especificación y diseño de los automatismos secuenciales. El procedimiento de diseño debe incluir además la validación formal de propiedades (Murata, 1998), (Silva, 1982) que garanticen el correcto desempeño del automatismo, empleando para ello métodos como los Grafos de ocurrencia o los invariantes (Kurt, 1997). La validación de las estructuras generalizadas propuestas será presentada en un próximo trabajo. Posteriores evoluciones de las redes, tales como las redes jerárquicas y las redes de alto nivel permiten la modelación y utilización de técnicas de programación estructurada, las que facilitan enormemente las tareas de diseño y prueba de programas. Futuros desarrollos estarán orientados a la utilización de simuladores de PLC basados en Redes de Petri, a la validación dinámica de modelos en redes jerárquicas y a la modelación de estructuras de control lógico mediante CPN.

REFERENCIAS

Adamski, M. *SFC, Petri Nets and application*

specific logic controllers: IEEE, 1998.

David, R. *Petri nets and Grafset*. Ed. Prentice Hall, 1992.

Desrochers A. *Applications of Petri Nets in manufacturing systems*: IEEE Press, 1995.

Espeleta, J. *Automatic synthesis of CPN for the control of FMS*". IEEE Transactions on robotics and automation, 13, (3), 1997.

Feldmann, K. *Specification, design and implementation of logic controllers based on CPN models*. IEEE transactions on control systems technology. 7, (6), 1999.

Ferreiro, R. *Método para implementar automatismos secuenciales en instalaciones industriales*. Montajes e instalaciones, 1995.

Frey, G. *Automatic implementation of PN-based control algorithms on PLC*. Proceeding of the America Control Conference, 2000.

Jiang J. *RT synchronization of multiaxis high speed machines, from SFC specification to Petri Nets verification*. IEE proceeding control theory applications. 143, (2), 1996.

Kurt, J. *Coloured Petri nets*. Ed. Springer Verlag, 1997.

Lennartson, B. *Modeling, specification and controller synthesis for DES*. IEEE, 1998.4

Mink S., W. *Inyección de plásticos*. México, 1981.

Miyazawa, I. *Verification of the behavior of SFC based on its PN model*. IEEE, 1997.

Moreno, C.M., *Análisis y diseño de sistemas secuenciales*. Silicio 7, Año 5.

Murata, T. *Petri nets: properties, analysis and applications*. Proceeding IEEE, 77, (4), 1989.

Perez Q., J. *Generación automática de código de expresiones lógicas de un programa de control basado en Grafset*. Universidad de Oviedo, 2000.

Peterson, J. *Petri nets theory and the modeling of systems*. Ed. Prentice Hall, 1981.

Pollard, J. *The future of PLC programming*. Control engineering, Febrero 1995.

Siemens S.A. *Manual SIMATIC S7-300*. 2000.

Silva, M. *Las Redes de Petri en la automática y la informática*. Madrid, 1982.

Zhou, M. *Comparing ladder logic diagrams*

- and Petri nets for sequence controller design.*
IEEE transactions on industry applications. 41,
(6), 1994.
- Zhou, M. *Design of industrial automated systems
via relay ladder logic programming and Petri
nets.* IEEE transactions on systems, man and
cybernetics. 28, (1), 1998.
- Zhou, M. *Discrete event control design methods.*
13th IFAC Word Congress, 9, 1996.
- Zhou, M. *Petri nets in flexible and agil automation.*
NJIT, 1995.
- Zurawski, R. Y Zhou, M. *Petri nets and industrial
applications.* IEEE Transactions on industrial
electronics, 41, (6), 1994.