

Gladys Bastidas Gustin Ph.D.

Directora Centro de Investigaciones y Servicios
Institución Universitaria Tecnológica de Comfacauca
gbastidas@tecnologicocomfacauca.edu.co

Elio R. Avila Rodríguez
ETESA, Las Tunas, Cuba.
eavila@itu.tel.etsesa.cu

Israel Benítez Pina
Facultad de Ingeniería Eléctrica, Universidad de
Oriente, Santiago de Cuba, Cuba.
ibenitez@uo.edu.cu

Modelamiento con GHENeSys IEC 1131 Compatible

Resumen

El presente artículo trata un acercamiento reciente de modelado sobre redes de Petri (PN) como método formal de diseño, verificación, validación e implementación de programas de PLCs. En éste se define la red jerárquica extendida GHENeSys IEC 1131 Compatible y se detalla una metodología que comprende esta red como formalismo a emplear. Ésta red persigue adaptarse a las particularidades de los PLCs como dispositivos de automatización y control, así como ganar en simplicidad, generalidad y facilidades de aplicación al aprovechar las potencialidades que brinda y que están dadas en su capacidad de obtener modelos jerárquicos, sus posibilidades de representación, su carácter general, los métodos de análisis que permite aplicar y la fácil traducción del modelo obtenido a un programa resultante IEC 1131 compatible a través de simples reglas de analogías.

Palabras clave: Redes de Petri, controladores lógicos programables (PLCs), sistemas de eventos discretos (DES).

Introducción

El estudio formal de automatizaciones con Controladores Lógicos Programables (PLCs) resulta tema de creciente interés entre los estudiosos y prácticos de esta rama, hecho que está dado por las ventajas de las soluciones obtenidas a partir de éstos métodos en cuanto a seguridad, calidad, generalidad, tiempos y costos de ejecución, entre otros.

En este contexto, la síntesis de programas de PLCs basados en modelos desarrollados sobre redes de Petri (PN), resulta un caso particular dentro de la síntesis de controladores de Sistemas de Eventos Discretos (DES) (ver Fig.1).

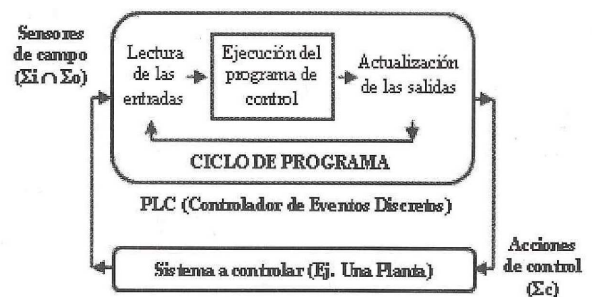


Fig.1 Sistema de eventos discretos (DES)

En la Teoría del Control Supervisorio de DES (Ramage and Wonham, 1989) se define un DES como “un sistema dinámico que involucra la posible ocurrencia de eventos físicos en intervalos irregulares desconocidos”. El control de DES asume que se puede impedir la ocurrencia de ciertos eventos del sistema cuando se desee, si se está prevenido de su

ocurrencia. Para modelar este control, se separa el conjunto total de eventos Σ , en eventos incontrolables y controlables ($\Sigma = i + c$). Los eventos c se pueden impedir en cualquier momento, mientras que los eventos i se modelan como aquellos que el agente controlador no puede influir sobre ellos.

Por otra parte, se separa el conjunto total de eventos en eventos observables, que son aquellos que se puede conocer su ocurrencia a través de sensores o cálculos, y los no observables, para los que no es posible conocerlo, es decir, $\Sigma = o + n$. De aquí se infiere que un evento incontrolable pueda ser a su vez observable.

Entre las especificaciones que se han considerado en el Control Supervisorio de DES están el problema de los estados prohibidos, que resultan especificaciones del control expresadas como condiciones prohibidas que deben ser evitadas y el problema de las cadenas prohibidas, en éstas las especificaciones se expresan como la ejecución de aquellas secuencias de actividades que no permitan la ocurrencia de las no deseadas. El controlador se sintetiza de forma tal que garantice que en el sistema no ocurran los estados prohibidos, transitando por los estados máximamente permisibles (Ramadge and Wonham, 1987).

Actualmente existe un gran número de trabajos que tratan el modelado sobre PN como método formal de síntesis de programas de PLCs, entre éstos han surgido diversas vertientes y tendencias, algunas de las cuales se apartan de las PN clásicas o tradicionales, creando extensiones o nuevas formas de interpretación de ellas (Holloway et al, 1997; Frey and Litz, 2000; Avila, 2002; Bani and Frey, 2003; Benítez et al, 2003; Uzam, 1998).

Aunque son indiscutibles los logros en este sentido, éstos métodos no han sido ampliamente aplicados en la industria (Mintchel, 2000), hecho dado principalmente por las limitaciones que aún existen en la correspondencia modelo-programa de PLC, así como por la imposibilidad de tratar fielmente particularidades de este equipamiento, como lo son: el ciclo de programa y su paralelismo inherente, la jerarquía (posible de obtener por

subrutinas o subprogramas), la diferenciación de las acciones de impulso y nivel, la manipulación de información no solo binaria, la adherencia a la norma internacional IEC 1131-3, entre otras. Además, los métodos desarrollados hasta el momento por una parte resultan complejos, mientras que por otra están restringidos a un lenguaje de programación específico o a un determinado método de análisis y verificación, no contando con la suficiente simplicidad, generalidad y facilidades de aplicación que les resulte atractivo a los profesionales de esta rama.

En este trabajo se presenta una red y una metodología de diseño formal que busca contribuir a la solución de las limitaciones antes mencionadas. El artículo se organiza de la siguiente forma: en la sección 2 se define formalmente uno de los últimos acercamientos de modelado en PN aplicado a los programas de PLCs, siendo este la red jerárquica extendida GHENeSys IEC 1131 Compatible. Posteriormente (sección 3) se presenta una metodología de diseño formal que emplea dicha red. Esta ha sido utilizada con buenos resultados en aplicaciones tanto docentes como industriales. Por último, se recogen las conclusiones y las referencias bibliográficas.

Red Jerárquica Extendida GHENeSys IEC 1131 Compatible.

La red **GHENeSys IEC 1131 Compatible** ha ido evolucionando paulatinamente y tiene sus orígenes en la red jerárquica extendida **GHENeSys** (General Hierarchical Enhanced Net System) (González and Silva, 2001). Es el resultado de expandir esta última al campo de aplicación de los PLCs y un acercamiento a la compatibilidad de las PNs y los lenguajes estandarizados en la norma internacional IEC 1131.

GHENeSys IEC 1131 Compatible se define como una quintupla $N = (P, T, F, M_o, Q)$, tal que:

- i). P es un conjunto finito y no vacío de lugares, con $P = P_c \cup P_n \cup P_m$, siendo P_c , P_n y P_m conjuntos dados.

- ii). T es un conjunto finito y no vacío de transiciones, con $T = T_c \cup T_u$, siendo T_c y T_u conjuntos dados. Por lo anterior se cumple que $P \cap T = \emptyset$. Debe satisfacerse además que $P \cap T = \emptyset$ y que para todo $t \in T$, los conjuntos $t^{(p)}$ y $t^{(m)}$ sean unitarios, con $p \in P_n \cup P_m$.
- iii). F es la relación de flujo y define un conjunto no vacío de arcos de peso unitario que interconectan los lugares a las transiciones y viceversa, y está compuesta por $F \subseteq \text{In}(P_c \times T) \cup \text{En}(P_c \times T) \cup (P_n \times T) \cup (T \times P_n) \cup (P_m \times T) \cup (T \times P_m) \cup \emptyset$.
- iv). M_0 es un marcaje inicial binario ($M(0) = 1$).
- v). Q es una función que asocia un conjunto finito de acciones de control a algunos lugares o al disparo de algunas transiciones, siendo $Q = Q_l \cup Q_p$, con $Q_l = \{O, \bar{O}\}$ y $Q_p = \{S, R\}$.

La red GHENeSys IEC 1131 Compatible es una red ordinaria que constituye una extensión de aquellas definidas como Condición/Evento (C/E) y cuenta con dos tipos de nodos, los lugares (representados por círculos), que pueden estar marcados o no (puntos negros) y las transiciones (representadas por barras o rectángulos). Ambos nodos se interconectan a través de los arcos, que pueden ser de tres tipos: inhibidores, habilitadores u ordinarios.

Según el inciso i) los lugares pueden ser de tres tipos, lugares de control ($P_c, P_c \cup P$), lugares normales o comunes ($P_n, P_n \cup P$) y los lugares "macros" o macrolugares ($P_m, P_m \cup P$), éstos últimos se representan por círculos de mayor tamaño para identificarlos del resto de los lugares. Seguidamente definimos su significado:

- Los P_c (llamados en GHENeSys Pseudoboxes) son lugares de marcaje persistente, pues no lo pierden con el disparo de las transiciones, se identifican con las letras PS y pueden representar:
 - o Información que brindan los sensores del proceso (bits asociados a las entradas del PLC, ejemplo pulsadores, microswitch, etc.).

- o Información de las señales de salida del PLC (bits relacionados con la activación de los actuadores o dispositivos de control).
- o Información asociada a otras subredes que representan variables globales del sistema (dependencia funcional) u otras partes de la propia red (se corresponden con bits internos del PLC, conocidos también como "banderas").
- o Los bits de salida de temporizadores y contadores.
- o Las salidas binarias de otros bloques funcionales (Ej. comparación, suma, resta, etc.).
- Los P_n se identifican con la letra P y representan estados por los que transita el sistema o tareas a ejecutar por este.
- Los P_m se identifican con las letras MP y constituyen un caso especial de lugar al agrupar o encapsular varias operaciones u otras partes de la red (subredes). Tienen un tratamiento similar a los bloques funcionales del lenguaje LD, y pueden representar:
 - o Llamadas a módulos o subredes de menor jerarquía, relacionados típicamente con tareas independientes, subprogramas o subrutinas.
 - o La activación de temporizadores, contadores y la ejecución de los demás bloques funcionales del PLC (Ej. comparación, suma, movimiento de datos, PID, etc.).

Para una transición $t \in T$, se define el conjunto de lugares de entrada o precondiciones (lugares de salida o postcondiciones) de t como $t^{(p)} := p \in P \mid (p,t) \in F$ ($t^{(m)} := p \in P \mid (t,p) \in F$, con $p \in P_n \cup P_m$). De igual forma, para un lugar $p \in P_n \cup P_m$, el conjunto de transiciones de entrada de p resulta $t^{(p)} := t \in T \mid (t,p) \in F$; para el caso de las transiciones de salida de los lugares abarca todos los subconjuntos que los integran, es decir, para $p \in P$ el conjunto de transiciones de salida de p resulta $p^{(t)} := t \in T \mid (p,t) \in F$. En palabras, los lugares comunes y los macrolugares pueden interconectarse tanto a la entrada como a la salida de las transiciones, y según el punto iii), se conectan

a éstas por medio de arcos ordinarios ($(P_n \times T) (T \times P_n) F$). Por otra parte, los lugares de control solo pueden ser lugares de entrada a las transiciones, y tomando en consideración el punto iii), se conectan a éstas solamente a través de arcos habilitadores ($En(P_c \times T) F$) o inhibidores ($In(P_c \times T) F$).

Las transiciones están divididas en dos subconjuntos, según el inciso ii) pueden ser transiciones controladas (T_c) o no controladas (T_u). Las controladas son aquellas cuyo conjunto de lugares de control de entrada sea no vacío, es decir, $T_c := \{t \in T \mid T^{(p)}_t \neq \emptyset\}$, mientras que las no controladas son aquellas cuyos lugares de entrada solo sean lugares normales o macrolugares ($T_u := \{t \in T \mid T^{(p)}_t = \emptyset\}$).

Para que una transición $t \in T$ esté habilitada todas sus precondiciones deben ser “verdaderas”, para ello se requiere el cumplimiento de las siguientes reglas:

- Para $t \in T_c$, con $p \in P_n$, $P_m y p = 1$ $M(p) = 1$.
- $t \in T_c$, con $p \in P_c$ y $En(p \times t) F$ $M(p) = 1$.
- $t \in T_c$, con $p \in P_c$ y $In(p \times t) F$ $M(p) = 0$.

El disparo de una transición $t \in T$ provocará:

1. La eliminación del marcaje en su lugar de entrada que no sea de control (para $p \in T^{(p)}_t$ y $p \in P_n$, P_m $M(p) = 0$).
2. El depósito de una marca en su lugar de salida (para aquel $p \in T^{(p)}_t$ $M(p) = 1$).
3. La ejecución de aquellas acciones de impulso asociadas a t inmediatamente después de su disparo ($Q_p(t)$).
4. La ejecución de aquellas acciones de nivel asociadas al lugar de salida, si este es un lugar común ($Q_i(p)$, con $p \in P_n$), o el bloque funcional modelado por un macrolugar (P_m), si fuera este el caso, al instante en que se deposita la marca en él.

El inciso v) comprende la función Q , que contempla los dos tipos de acciones posibles de utilizar en los PLC (de impulso o de nivel),

recogidos en los puntos 3 y 4 de las reglas de disparo. Los valores que Q puede tomar son: “0” (desactivado) ó “1” (activado). Como ya se ha visto, el disparo de una transición $t \in T$ ejecuta aquellas acciones de impulso (instrucciones del tipo SET o RESET del PLC, relacionadas con los subconjuntos S y R por ese orden) asociadas a ella ($Q_p(t) = S(t) R(t)$), inmediatamente después de su disparo. Las acciones de nivel (instrucciones del tipo asignación y asignación negada, correspondiéndose respectivamente con los subconjuntos O y \bar{O}), se ejecutarán en el momento que se deposita la marca en el lugar $p \in P_n$ asociado a ellas ($Q_i(p) = O(p) \bar{O}(p)$), y permanecerán activadas mientras p se encuentre marcado (mientras $M(p) = 1$, con $p \in P_n$ $Q_i(p)$ estarán activadas). Es posible definir tantas acciones como se requieran en cada transición o lugar asociado a ellas.

Los dos primeros puntos de las reglas de disparo implican por sí mismos acciones sobre el proceso, al modificar el estado de este. Posteriormente, en la parte de implementación del modelo obtenido sobre GHENeSys IEC 1131 Compatible se verá cómo este cambio de estados se resuelve con acciones de impulso, con un tratamiento idéntico al visto anteriormente.

Partiendo de las definiciones anteriores, podemos denominar al **modelo no controlado** (N_u) de N a aquel conformado solamente por lugares comunes, transiciones no controladas y arcos comunes que los interconectan (no incorporan los lugares de control, los macrolugares ni las acciones), es decir, es aquel $N_u \subseteq N$ tal que $N_u = (P_n, T_u, F_u, M_u)$, siendo $F_u = (P_n \times T) (T \times P_n)$. De esta forma se establece un modelo que comprenderá solamente aquellos estados por los que transita el sistema o tareas a ejecutar. Por otra parte, el **modelo controlado** (N_c) es aquel que hace uso de todos los elementos de GHENeSys IEC 1131 Compatible, incluyendo los lugares de control y las acciones ($N_c = N$).

Del inciso ii) es posible deducir una particularidad que introduce GHENeSys IEC 1131 Compatible, al quedar restringido su modelo no controlado a

redes clasificadas como Máquinas de Estado (SM) (Murata, 1989), es decir, $t \in T$ y $p \in P_n$ $\forall t \in T$. De esta forma es posible conformar el espacio total de estados del sistema utilizando estructuras condicionales (de decisión o conflictos). A los efectos del modelo esto ocurre cuando varias transiciones comparten el mismo lugar de entrada (forman un conjunto $C := \{t \in T, \text{ con } p \in P_n \text{ y } p = 1\}$, siendo su cardinal $C \geq 2$). Si tenemos en cuenta lo anterior, se puede plantear entonces que esta clasificación es suficiente para modelar cualquier programa de PLC que se desee por este método, teniendo en cuenta que éstos son posibles de construir a partir de estructuras IF-THEN-ELSE anidadas, según los requerimientos del sistema a controlar.

Por otra parte, la restricción a SM permite la obtención de un modelo que se asemeja mucho a un programa real de PLCs, al simular éstos últimos el paralelismo en la ejecución de tareas a partir de su ciclo de programa, en este se incluye la ejecución secuencial del programa de control y la actualización de sus salidas una vez que el mismo ha sido recorrido totalmente. Este es un aspecto que requiere ser conocido y considerado por los programadores a la hora de concebir sus soluciones, pues de no tenerse en cuenta este hecho pueden no obtenerse los resultados esperados. Siguiendo éstos principios, para la implementación final del modelo desarrollado sobre GHENeSys IEC 1131 Compatible, se asume la secuencia de ejecución de arriba a abajo y de izquierda a derecha, de forma similar a como se hace para el lenguaje LD.

Una propiedad importante de todo programa de PLC es su determinismo. Esto significa que debe responder de igual manera ante las mismas condiciones de operación (Ej. encontrándose en un estado determinado, para cada secuencia de entrada debe responder con idéntica secuencia de salida). Si esto no se cumple, entonces el controlador tendrá una respuesta inespecífica en determinadas situaciones de operación, dependiendo para su correcto funcionamiento del

posible estado aleatorio de ciertos elementos de su implementación.

En un modelo sobre GHENeSys IEC 1131 Compatible, al tener un marcaje inicial binario ($p \in P \implies M(p) = 1$) y la estructura de su modelo no controlado definida como SM, se satisface el determinismo si en el modelo controlado, para cada estructura condicional, sus condiciones asociadas son excluyentes. Esto garantiza que el disparo de una transición no dependa del marcaje de su postcondición, pues solamente estará marcado un lugar común o macrolugar a la vez ($P_i, P_j \in P_n \implies M(P_i) + M(P_j) = 1$). Además, esto posibilita la obtención de redes seguras desde la propia etapa de formalización y facilita a su vez la implementación posterior a su programa en código de PLC.

En resumen, en GHENeSys IEC 1131 Compatible los lugares normales representan estados por los cuales transita el sistema o tareas a ejecutar por este, y los eventos y bits internos del PLC (banderas) se relacionan con los lugares de control. La dinámica del sistema se representa por medio de la presencia de marcas en los lugares, éstas indican el estado en que se encuentra el sistema modelado en ese preciso momento. Las transiciones son componentes activas que modelan acciones, éstas últimas ocurren al dispararse las transiciones (si se encuentran habilitadas). Como consecuencia de ello, cambia el estado del sistema (modificando así el marcaje de la red) y se ejecutan las acciones asociadas a la transición disparada y a su lugar de salida, o la ejecución de los bloques funcionales asociados a su macrolugar de salida, si este fuera el caso. Las transiciones se disparan si se cumplen todas sus precondiciones. Esto provoca la eliminación de las marcas de los lugares de entrada y su adición a los lugares de salida, exceptuando los lugares de control, donde no cambia el estado del marcaje con el disparo de las transiciones, a no ser que con este se modifique el estado de la información que los mismos modelan. En la Fig. 2 se muestran los elementos que componen a la red GHENeSys IEC 1131 Compatible y el disparo de una transición.

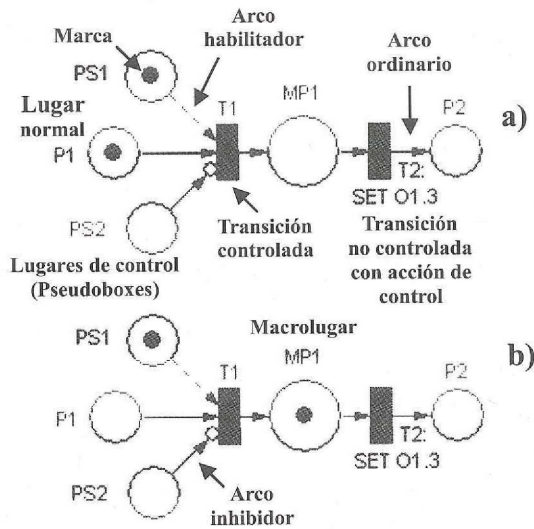


Fig. 2 Elementos de GHENesys IEC 1131 Compatible y disparo de una transición a) antes del disparo b) Después del disparo.

Tratamiento de las subredes

Una subred es una abstracción que se utiliza para modelar una parte de la red que por razones de simplicidad, transparencia o conveniencia se sustituye por un elemento de red que la identifica, pudiendo sustituirse tanto por transiciones como por lugares. En el caso de la red que presentamos, esto se reduce solo al empleo de lugares (macrolugares), que al poder permanecer marcados cierto tiempo se adaptan al caso real de operación del PLC, donde una subrutina puede tardar un tiempo determinado en ejecutarse; no así las transiciones, que una vez que están habilitadas su disparo es inmediato. Además, se persigue de esta forma darle un tratamiento similar al de los bloques funcionales que son posibles de utilizar en los PLCs (entre los que se incluyen las llamadas a subrutinas o subprogramas), empleándose de la misma manera en que están concebidos éstos.

Una subred (SN) en GHENesys IEC 1131 Compatible se construye con los mismos elementos de red que han sido definidos anteriormente (SN N), y particularmente comprende un lugar común de inicio y otro de fin del elemento que modela (SN N (P_{inicio}, y P_{fin}) P_n). Las subredes pueden ser reversibles o no. Las

reversibles (SN_{rev}), son aquellas que cuentan con una transición entre sus lugares comunes de inicio y fin (SN_{rev} := {SN N en las que t T, tal que P_{fin}^(t) = ^(t)P_{inicio}}). Comúnmente, éstas modelan elementos del sistema que han hecho un ciclo completo de su ejecución y se encuentran nuevamente en su estado inicial de operación. Un ejemplo de esto es aquella subred que encierra el ciclo de arranque-parada de un motor, una vez efectuado el arranque, la parada lo sitúa nuevamente en su estado inicial, a la espera de otra orden de arranque para su operación (ver Fig. 3).

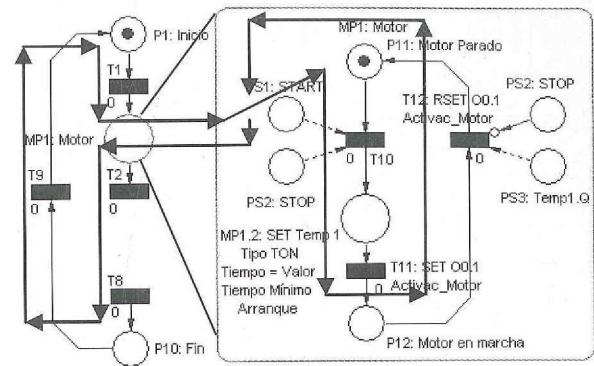


Fig. 3 Secuencia de ejecución GHENesys IEC 1131 compatible atendiendo a su objetivo jerárquico

Toda subred de menor jerarquía que se encuentre encapsulada dentro de una que es reversible, lo será a su vez, esto se cumple además para las subredes anidadas. El planteamiento anterior se puede inferir de la propia definición de subred, pues al ser una parte de la red que se reemplaza por las razones ya expuestas, hereda entonces esta propiedad de la red a la que pertenece (SN SN_{rev} N t T_u, tal que P_{fin}^(t) = ^(t)P_{inicio}).

El marcaje de un macrolugar en un módulo de mayor jerarquía modela la llamada y ejecución del módulo de menor jerarquía asociado a dicho macrolugar (si M(p) = 1, con p P_m SN(p) se ejecutará). Al concluir este módulo la ejecución continuará en la transición inmediata posterior al macrolugar en que se quedara (ver Fig. 3). Se pueden establecer tantos niveles jerárquicos como se deseen, no teniendo que coincidir necesariamente con los detalles de implementación. Para denotar la jerarquía se propone el empleo de subíndices, los que permiten

indicar el nivel jerárquico de un macrolugar, y en caso de anidamiento, representar a cuál macrolugar de mayor jerarquía se “subordina”, es decir, al que se relaciona.

De la misma forma en que es posible hacer una llamada a subrutina en cualquier parte de un programa de PLC y las veces que se desee hacerlo, en GHENeSys IEC 1131 Compatible es posible utilizar aquellos macrolugares que modelan las llamadas a subrutina a conveniencia del diseñador y bajo los mismos principios que en un PLC.

Los macrolugares permiten la obtención de un modelo más refinado jerárquicamente, al poder establecer un programa principal descomponiéndolo en módulos (subredes) de menor jerarquía con independencia entre ellos, relacionándose solamente por las variables comunes (globales). Esto se adhiere al diseño “descendente-ascendente” (Top-down–Bottom-up) propuesto por varios autores al descomponer el modelo en subredes tan pequeñas como se desee (Zhou, 1995). Lo anterior evita el problema de la explosión de estados al poder analizar las subredes por separado con mayor facilidad. El resultado global del sistema se obtiene unificando los resultados parciales de cada subred analizada. Si toda $SN \ N$ es viva, segura y determinística, entonces N lo será.

Tratamiento de los temporizadores, contadores y otros bloques funcionales.

Como ya se ha visto, los macrolugares se utilizan no solo para modelar las llamadas y ejecución de las subrutinas, sino que agrupan también los demás bloques funcionales que son posibles de utilizar en la programación de los PLCs.

Para el caso específico de los temporizadores, contadores, funciones de comparación y otras con tratamiento similar a éstas, a los efectos del programador, usualmente interesan solo los bits asociados a ellas una vez que éstas se han activado (los que se modifican como resultado de su ejecución), siendo “transparente” al programador su funcionamiento y operaciones internas, como

lo son: tiempos, conteos, etc. Por ello, normalmente se utiliza la información deseada a partir de los bits asociados a los bloques funcionales, como lo pueden ser aquellos que indican la terminación del tiempo o el completamiento del conteo programado, los de mayor o menor que, igualdad, acarreo, etc., según el caso que se requiera. En la Fig. 3 se muestra, de forma ilustrativa, la activación de un temporizador TON y el chequeo de su bit asociado que indica la culminación del tiempo programado.

Correspondencia con la norma internacional IEC 1131-3.

El nombre de GHENeSys IEC 1131 Compatible indica la correspondencia entre el modelo obtenido sobre esta red y los lenguajes de programación recogidos en la norma internacionalmente aceptada IEC 1131-3 a través de reglas generales de analogías. Esto permite definir equivalencias directas entre las estructuras básicas de los modelos y las secciones de programas IEC 1131 compatibles que resultan de su traducción.

Las reglas de analogías se relacionan seguidamente:

1. El estado 0 ó 1 de los lugares comunes y los lugares de control se corresponde con el estado de una variable binaria interna del PLC (marca de bit o bandera).
 - Si representan mediciones desde sensores del proceso, salidas o informaciones asociadas a otras subredes u otras partes de la propia red, según la IEC 1131 pueden utilizar un identificador simple o el direccionado directo del PLC.
 - Si representan el bit de salida de un temporizador o contador, se emplea el identificador del temporizador o contador seguido de “.Q”.
 - Si representan la salida binaria de algún bloque funcional, tendrán el nombre del bloque funcional seguido de “.nombre de la salida del bloque” como reglamenta la IEC 1131-3.
2. El estado 0 ó 1 de los macrolugares se corresponde con la ejecución de un bloque funcional del PLC.

3. Las acciones de impulso o nivel modifican recursos internos del PLC, ya sean bits relacionados con señales de salida o banderas definidas por el programador, por lo que utilizan el mismo identificador del bit al que se asocian, según lo indica la IEC 1131-3.

Metodología de Diseño Formal de Programas de PLCs Utilizando GHENeSys IEC 1131 compatible.

En el proceso de diseño de programas de PLCs proponemos los siguientes pasos:

1. Estudio de las particularidades y los requerimientos funcionales del sistema a controlar.
2. Definición del equipamiento que garantice la operación correcta del sistema de control, aunque pudiera modificarse este como resultado del propio proceso de diseño y análisis ulterior.
3. Determinación de la mayor cantidad de unidades funcionales del sistema a controlar, sin coincidir necesariamente con su futura implementación.
4. Definición preliminar de las acciones internas e interdependencias de cada unidad funcional y el nivel jerárquico en que deben estar de acuerdo a las interdependencias y a su función en el sistema de control.
5. Definición de un modelo jerárquico estableciendo una red GHENeSys IEC 1131 Compatible para todo el sistema de control, aplicando un diseño “descendente” (Top-down) (Zhou, 1995) que abarque las unidades funcionales definidas previamente.
6. Refinamiento “ascendente” (Botton-up) (Zhou, 1995) del modelo para depurar las relaciones internas de cada subred y las interdependencias entre ellas.
7. Determinación de las principales propiedades de cada subred definida (fundamentalmente vivacidad, seguridad y reversibilidad). Se propone la aplicación del método de las reglas de reducción simples (Murata, 1989; Desel and Esparza, 1995) sobre el modelo no controlado, aunque no está limitado solo a este. Se comprueba además su determinismo verificándolo sobre el modelo controlado.

8. Modificación de cada subred, realizando oportunas adecuaciones que garanticen el cumplimiento de las propiedades antes mencionadas.

9. Simulación del trabajo de cada subred utilizando una herramienta asistida por computadoras y comprobación del cumplimiento de los requisitos funcionales del usuario, modificándolas convenientemente si no cumpliera con ellas.

10. Determinación final de las propiedades de cada subred modificada y del cumplimiento de los requisitos funcionales del usuario. Repetición de los pasos anteriores mientras el resultado sea falso.

11. Traducción del modelo finalmente obtenido a un lenguaje de programación que soporte el PLC escogido. Se propone principalmente el SFC a todos los niveles definidos, quedando a criterio del usuario la selección de otro tipo de lenguaje (ST, IL, LD) para redes básicas.

Etapa de especificaciones informales

Esta etapa constituye el primer aspecto a considerar en el proceso de diseño (pasos 1 y 2 de la metodología), que abarca el estudio del proceso a controlar, sus requerimientos y la definición de la estructura de control, sensores de campo, entre otros, que garanticen la operación correcta del sistema de control.

Etapa de formalización

Esta etapa agrupa los pasos del 3 al 6 de la metodología presentada y tiene como objetivo la obtención de un modelo finito del sistema que se analiza, fragmentado este en las menores unidades funcionales posibles. Para ello se requieren definir los niveles jerárquicos en la estructura del modelo e identificar las interrelaciones entre los módulos (subredes) que lo forman, utilizando una red jerárquica extendida GHENeSys IEC 1131 Compatible.

En esta etapa se adapta el método Controlled Token Passing Marking (**C-TPM**) (Uzam, 1998). En él se crea primeramente un modelo no controlado del sistema a partir de las reglas TPM,

obtenidas directamente de sus especificaciones de estados prohibidos. Posteriormente, se obtiene el modelo controlado gobernando convenientemente el disparo de aquellas transiciones (cadenas deseadas) que garanticen el tránsito del sistema solamente por los estados máximamente permisibles. Para ello se incorporan los lugares de control, que se conectan a aquellas transiciones controlables por medio de arcos habilitadores o inhibidores, según se requiera.

Las reglas TPM no son más que estructuras del tipo:

```
If <marcaje(s)>
Then <una transición controlable estará
habilitada>
```

La parte *If* de la regla TPM comprende uno o varios marcajes del modelo, mientras que la parte *Then* indica la habilitación y disparo de una transición controlable si se cumplen todos los marcajes contemplados en la parte *If*, de lo contrario estará bloqueada. Cuando se contemple más de un marcaje en la parte *If*, se requiere el uso de las funciones lógicas AND y OR para relacionarlos. En este último caso (función OR), la transición controlable de la parte *Then* se duplicará $2^n - 1$ veces, siendo n el número de lugares que comprende el marcaje.

El modelo no controlado se construye utilizando las estructuras básicas de secuencia, conflicto y atribución, teniendo en cuenta su restricción a redes clasificadas como SM, como se ha mencionado anteriormente.

Etapa de verificación

La verificación es la prueba de que la semántica interna del modelo sea correcta, independientemente del sistema modelado. En esta etapa se comprueba que el algoritmo de control cumpla propiedades funcionales estándares, principalmente vivacidad, seguridad y reversibilidad. A estas propiedades específicas de las PN se adiciona el chequeo del determinismo del modelo obtenido.

En esta etapa de diseño se realizan las modificaciones que sean necesarias sobre el **modelo no controlado** hasta lograr redes vivas y seguras. Se realiza de esta forma tomando en consideración que formalmente no existe diferencia entre el modelo controlado y el no controlado, pues en ambos se recoge el comportamiento deseado del sistema a controlar y abarcan el espacio total de estados del sistema modelado ($P_n N_c = P N_u$).

En GHENeSys IEC 1131 se simplifica esta etapa al poder realizar la verificación de las propiedades a partir del **método de las técnicas de reducción** (Murata, 1989; Desel and Esparza, 1995). Esto es posible, al estar conformado el modelo obtenido por redes **puras** (Murata, 1989). El modelo queda libre de autolazos al contener al menos dos lugares y una transición (uno de inicio y otro de fin) y estar restringido a redes SM, por ello $t T y P^{(t)} t^{(p)}$.

Este método facilita el análisis de sistemas “grandes” al reducir estructuralmente el modelo obtenido a una descripción más general, denominada **subred o “macro”**, la cual mantiene las propiedades originales de la red que le dio origen. En (Murata, 1989) se muestran algunas de las transformaciones que se pueden realizar, siendo las más utilizadas para nuestro caso la fusión de nodos (lugares o transiciones) serie y la fusión de nodos (lugares o transiciones) paralelos.

Los aspectos de formalización definidos para GHENeSys IEC 1131 Compatible garantizan que una red sea **fuertemente conectada** (Murata, 1989) si esta es reversible, pues para dos nodos cualquiera de la red existirán caminos directos que los interconecten.

Una red resulta **estructuralmente correcta** si es posible reducirla hasta obtener una **red mínima**, es decir, que cuente solamente con un lugar y una transición ($SN_{min} := \{SN N \text{ tal que } P=T=1\}$). Si esto se logra en GHENeSys IEC 1131 Compatible es porque la red resulta además reversible ($SN_{min} N SN_{rev} N$), y por tanto es fuertemente conectada. Por lo referido anteriormente, es posible comprobar las propiedades de vivacidad y

seguridad aplicando los teoremas de (Murata, 1989) para redes SM (como ha sido definida la que presentamos). Al poseer un marcaje inicial la red será **viva**, y **segura** al contar este solo con una marca (recordar que $M(0) = 1$), es decir, el marcaje inicial binario resulta condición necesaria y suficiente para que toda SN_{\min} sea viva y segura.

El determinismo se comprueba verificando sobre el **modelo controlado** que toda estructura condicional posea condiciones disjuntas.

Al descomponer el sistema en pequeñas subredes, la verificación se realiza de forma simple y prácticamente por simple inspección.

En (Avila et al, 2004) se ejemplifica el proceso de verificación sobre GHENeSys IEC 1131 Compatible y se muestra cómo no está limitado solamente al método gráfico de las técnicas de reducción. Una variante de acercamiento por ecuación matricial es el **Teorema del Rango** (Desel and Esparza, 1995). Este teorema brinda una vía de verificación analítica que permite una solución completa al problema de definir vivacidad y seguridad, por la vía polinomial a todo el tamaño de la red y a través de la caracterización de redes de Libre Selección (FC-nets) bien-formadas (Murata, 1989). Por ello, es posible aplicarlo al caso que nos ocupa de redes SM, al ser una clasificación más baja, que está contenida dentro de las FC-nets.

De acuerdo al Teorema 6.14 de (Desel and Esparza, 1995), esto puede ser resuelto mediante los siguientes pasos:

- 1°. Debe comprobarse que la red sea bien-formada, para eso debe cumplir los 4 puntos del Teorema 6.14. Visualmente se puede comprobar que una red esté conectada y tenga al menos un lugar y una transición.
- 2°. La existencia de un invariante S positivo, lo que se logra resolviendo el sistema de ecuaciones lineales:

$$A \cdot Y = \emptyset, \text{ siendo } Y \quad (1, \dots, 1)$$
- 3°. La existencia de un invariante T positivo, esto se logra resolviendo el sistema de ecuaciones lineales:

$$A^T \cdot X = \emptyset, \text{ siendo } X \quad (1, \dots, 1)$$

4°. Luego se realiza el conteo de clusters.

En (Desel and Esparza, 1995) se define un cluster como sigue: Sea x un nodo de una red. El cluster de x , denotado por $[x]$, es el mínimo conjunto de nodos tal que:

- $x \in [x]$,
- si el lugar p pertenece a x entonces $p^{(i)}$ se incluye en $[x]$, y
- si la transición t pertenece a x entonces $t^{(i)}$ se incluye en $[x]$.

Con el valor de este conteo de clusters (C_n) y el cálculo del rango de la matriz de incidencia A de la red, se determina si $\text{Rank}(A) = |C_n| - 1$.

Etapa de validación

La validación es la etapa en que se determina si el modelo cumple con el propósito del diseñador. En este caso se relaciona con los pasos 9 y 10 de la metodología. Para ello se requiere simular el trabajo de cada subred con la ayuda de un asistente por computadoras. Aun no existe la herramienta óptima que soporte todo el fundamento de GHENeSys IEC 1131 Compatible. Por ello es necesario utilizar, de forma alternativa, alguno de los programas que se acerquen a este. En Internet se encuentran disponibles sin costo alguno el Visual Object Net ++ y el HSim, los que incorporan lugares de control y arcos habilitadores e inhibidores. De esta forma se puede comprobar que el funcionamiento simulado de cada subred modelada se corresponda con las exigencias del sistema, las cuales fueron recogidas en la etapa inicial de diseño.

Etapa de implementación

El último paso de la metodología (paso 11) se relaciona con la etapa de implementación, y tiene como objetivo lograr la traducción del modelo obtenido a su equivalente en código de programa PLC, cumpliendo para ello las exigencias del lenguaje IEC 1131 compatible que admita el equipamiento escogido. Esta traducción se realiza siguiendo reglas que se han conformado de incorporar las acciones a las C-TPM tratadas en el punto 3.1, teniendo en cuenta para ello el cumplimiento de las reglas de disparo de esta red.

El programa se conforma combinando las estructuras típicas AND y OR que han sido definidas para el modelo. Para una estructura AND como se muestra en la Fig. 4 a) quedará de la siguiente forma:

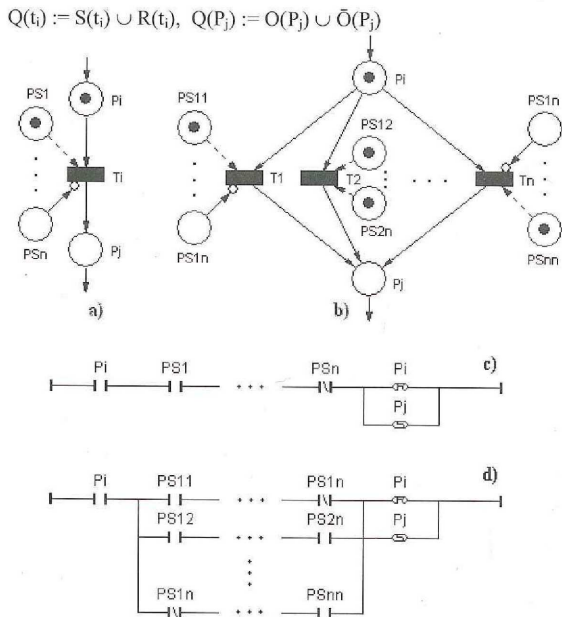


Fig. 4 Estructuras típicas de GHENeSys IEC 1131 Compatible a) AND b) combinación AND-OR c) equivalencia del AND en LD d) equivalencia de la combinación AND-OR en LD

If $M(P_j) = 1$ and $M(P_{en}) = 1$ and $M(P_{in}) = 0$,
 con $P_{en} := \{P_c^{(P)} t_i \text{ En}(P_{e,t_i}) F\}$ y $P_{in} := \{P_c^{(P)} t_i \text{ In}(P_{e,t_i}) F\}$
 Then RESET[(P) and R(t)] y SET[(P) and S(t)]

If $M(P_j) = 1$, con P_j, P_n, P_m
 Then [OUT(O(P))] and [neg_OUT(O(P))], para P_j, P_n . Si P_j, P_m se ejecuta la función que modela el macrolugar.

Para una combinación AND-OR del tipo mostrado en la Fig. 4 b) resulta como sigue:

If $M(P_j) = 1$ and [(M(P_{en1}) = 1 and M(P_{in1}) = 0) or (M(P_{en2}) = 1 and M(P_{in2}) = 0) or (M(P_{enn}) = 1 and M(P_{inn}) = 0)] con
 $P_{en1} := \{P_c^{(P)} t_i \text{ En}(P_{e,t_i}) F\}$ y $P_{in1} := \{P_c^{(P)} t_i \text{ In}(P_{e,t_i}) F\}$ y $i = 1, 2, \dots, n$
 Then RESET[(P) and R(t)] y

SET[(P) and S(t)], cumpliendo que:

$$t_1 = t_2 = \dots = t_n; \text{ por tanto,}$$

$$R(t_1) = R(t_2) = \dots = R(t_n) \text{ y}$$

$$S(t_1) = S(t_2) = \dots = S(t_n)$$

If $M(P_j) = 1$

Then [OUT(O(P))] and [neg_OUT(O(P))], para P_j, P_n . Si P_j, P_m se ejecuta la función que modela el macrolugar.

Uno de los inconvenientes de la síntesis de programas de PLCs a partir de métodos formales de diseño, lo constituye el hecho de que puede generar en ocasiones mayor código de programa que el obtenido por la implementación directa, inconveniente que se justifica y compensa plenamente con las múltiples ventajas que introducen éstos. Por lo anterior, se recomienda realizar finalmente una depuración y eliminación de variables redundantes.

Conclusiones y Trabajos Futuros

En este trabajo se ha presentado un acercamiento de modelado sobre redes de Petri, aplicado al caso específico de las soluciones con PLCs. El mismo facilita la aplicación de este método formal de diseño, como aspecto importante para lograr su amplia utilización por los estudiosos y prácticos de esta rama de la automática, y con ello, aportar en la obtención de soluciones más rápidas, seguras, eficientes, de mayor calidad, transparencia y posibilidades de reutilización.

El trabajo se orienta a lograr una mayor correspondencia entre el modelo obtenido y su programa resultante en código PLC, permitiendo modelar las particularidades más importantes que soporta este equipamiento en las condiciones contemporáneas, sus potencialidades y su correspondencia con la norma internacionalmente aceptada IEC 1131-3. De la misma forma, simplifica el proceso de formalización, verificación, validación e implementación posterior. La restricción a redes Máquina de Estados (SM), lejos de resultar dañino, contribuye a este propósito y garantiza totalmente las exigencias a cumplir por los sistemas desarrollados sobre esta base.

Como trabajos futuros se prevé el desarrollo de una herramienta computacional que soporte los requerimientos de GHENeSys IEC 1131 Compatible y permita la traducción automática del modelo editado, verificado y validado sobre esta red, a su programa correspondiente de PLC.

Bibliografía

- ARIAS, K., VILLAFRUELA, L., BENÍTEZ, I. y OBREGÓN, O. (2003). "Sistema Docente de Diseño Formal de Automatizaciones". Memorias 1^{er} Taller Internacional sobre Control por Computadoras de Sistemas Industriales: Teoría y Aplicaciones. INFO'2003. Marzo.
- AVILA, E. (2002). "Modelado en PNs de Automatizaciones con PLCs de la Industria Azucarera". Tesis en opción del Título de Master en Automática. Departamento de Control Automático. Facultad de Ingeniería Eléctrica. Universidad de Oriente. Santiago de Cuba. Julio.
- AVILA, E., BENÍTEZ, I., y SILVA, J. R. (2004). "Síntesis de Programas de PLCs Modelados sobre GHENeSys IEC 1131 Compatible. Un caso de Estudio en la Industria Cubana". Memorias Conferencia Internacional TELECOM'2004. Santiago de Cuba. Julio.
- BENÍTEZ, I., AVILA, E., SILVA, J.R. y VILLAFRUELA, L. (2003). "Utilización de las Redes de Petri como Método Formal de Diseño de Automatizaciones con PLCs". Memorias 1^{er} Taller Internacional sobre Control por Computadoras de Sistemas Industriales: Teoría y Aplicaciones. INFO'2003. Marzo.
- DENIS, A. (2003). "Implementação de estrutura de controle de sistema a eventos discretos em controlador lógico programável utilizando a teoria Controle Supervisório Modular Local", Universidade Federal de Santa Catarina, Departamento de Automação e Sistemas, Junho.
- DESEL, J. y ESPARZA, J. (1995). "Free Choice Petri Nets". Cambridge University Press. Great Britain.
- FREY, G. y LITZ, L. (2000). "Formal Methods in PLC Programming". Proc. of the IEEE Conference on Systems Man and Cybernetics. SMC'2000. Nashville (USA). October 8-11.
- GONZÁLEZ, P. y SILVA J.R.(2001). "GHENeSys: Uma Rede Estendida para a Modelagem, Analise e Projeto de Sistemas Complexos". Proc. of SBAI'2001. Sao Paulo (Brasil). November.
- Survey of Petri Net Methods for Controlled Discrete Event Systems". Journal of Discrete Event Dynamic Systems. Vol. 7. No. 2
- MINTCHEL, G.A. (2000). "Power Up Programming with Graphical Modeling". Control Engineering. December. Vol 1.
- MURATA, T. (1989). "Petri Nets: Properties, Analysis and Applications". Proc. of IEEE. Vol. 77. No. 4. April.
- RAMADGE, P.J. y WONHAM, W.M. (1989) "The Control of Discrete Event Systems", Proceeding of the IEEE, Vol. 77, 1989, pp.81-97.
- RAMADGE, P.J. y WONHAM, W.M. (1987). "Supervisory Control of a Class of Discrete Event Processes". SIAM Journal of Control Optimization. Vol. 25. No. 1.
- UZAM, M. (1998). "Petri Net Based Supervisory Control of Discrete Event Systems and their Ladder Logic Diagram Implementations", Ph.D. Thesis, University of Salford, UK.
- ZHOU, M. (1995). "PN in Flexible and Agile Automation", Boston MA, Kluwer Academic Publisher.