

MODELOS DE GRAFCET Y APLICACIONES DISTRIBUIDAS EN LA NORMA IEC 61499. UN CASO DE ESTUDIO.

Oscar Miguel-Escrig, Julio-Ariel Romero-Pérez
Departamento de Ingeniería de Sistemas y Diseño. Universitat Jaume I
Av. Vicent Sos Baynat s/n. CP 12071 Castelló de la Plana. España
e-mail: {omiguel,romeroj}@uji.es

Resumen

En este trabajo se presenta un caso de estudio en el que se diseña una aplicación mediante distintos modelos en Grafcet los cuales van a ser implementados según el estándar para el diseño de aplicaciones distribuidas IEC 61499. Los diseños planteados tienen en cuenta distintos grados de modularidad de los componentes y, en base a estos modelos, se plantean distintas implementaciones, viendo cómo los modelos influyen en la distribución de los algoritmos de control.

Palabras clave: Grafcet, IEC 61499, DCS

1. INTRODUCCIÓN

En la actualidad la mayor parte de los sistemas de automatización industrial se programan según el estándar IEC 61131 [5], ampliamente usado en la programación de Automatas Programables Industriales (APIs). Sin embargo, en los últimos años se ha estado desarrollando el estándar IEC 61499 [3, 10], más orientado a la programación de sistemas de control distribuidos y reconfigurables. Dicho estándar permite el diseño centralizado de una aplicación de control cuya ejecución puede ser distribuida entre distintos dispositivos de control los cuales no necesariamente deben ser del mismo fabricante. Esto mejora de forma considerable la interoperabilidad del software de control. Otra de las diferencias respecto al IEC 61131 es que el nuevo estándar cuenta con mecanismos para la modificación de las aplicaciones sin necesidad de detener su ejecución, facilitando la reconfiguración de los sistemas de control.

Las características antes mencionadas hacen que el IEC 61499 esté siendo actualmente el centro de atención de numerosos trabajos para su uso en diferentes problemas de automatización y control industrial. Muestra del interés que entre los fabricantes de los equipos de control está suscitando el IEC 61499, es el hecho de que ya existen entornos de programación comerciales que lo soportan como Isagraf de Rockwell Automation, nxtSTUDIO de nxtCONTROL y EcoStruxure Automation Ex-

pert de Schneider Electric, que actualmente está en desarrollo.

Existen varios métodos de modelado del comportamiento de sistemas de automatización: redes de Petri interpretadas para control, diagramas de Grafcet, diagramas de estado UML, entre otros. Estos modelos permiten una descripción formal de las especificaciones del comportamiento de los sistemas, evitando de esta forma las ambigüedades a las que puede inducir una descripción textual (descripción informal) del comportamiento. Una de las características fundamentales de estos modelos es su grado de abstracción, lo cual permite modelar el comportamiento de los sistemas independientemente de la forma y la tecnología que se use en la implementación final de los controladores. Varios trabajos de investigación han abordado el desarrollo de metodologías para la traducción de los modelos a aplicaciones para Automatas Programables, teniendo en cuenta los distintos lenguajes de programación recogidos en la IEC-61131 (SFC, LD, FBD, ST, IL), [1, 6, 7, 9]. El objetivo de dichas metodologías es sistematizar y en última instancia automatizar el proceso de generación de los programas a partir de los modelos, reduciendo así las posibles fuentes de errores a la vez que se agiliza el proceso de programación.

Hasta el momento, muy pocos trabajos han abordado el desarrollo de metodologías que permitan el desarrollo de aplicaciones de control en el estándar IEC 61499. En particular, las ingenieras e ingenieros de control carecen de herramientas que permitan implementar, de forma sistemática, controladores para sistemas de eventos discretos a partir de las técnicas de modelado con las que están familiarizados y que utilizan para describir de una manera formal el comportamiento dinámico de los sistemas automatizados. Uno de los avances más recientes en este sentido ha sido presentados en [8] donde se propone una primera aproximación a la metodología para la implementación distribuida de modelos de Grafcet usando la IEC 61499.

En este trabajo se presenta un caso de estudio basado en una aplicación distribuida sencilla. A partir de un enunciado sobre la funcionalidad a implementar, se proponen varios modelos en Graf-

cet, el cual se define en el estándar IEC 60848 [4], que luego se implementan en la norma IEC 61499. Se analizará cómo el planteamiento de los modelos influye en modularidad y distribución del *software* de control que se obtiene a partir de los mismos.

El artículo se distribuye como sigue. En la Sección 2 se describe la plataforma experimental sobre la que se desarrollan las aplicaciones y se describe la funcionalidad a implementar. En la Sección 3 se presentan los diferentes modelos en Grafcet sobre los que se trabajará la implementación. En la Sección 4 se presentan las distintas implementaciones en la norma IEC 61499. Finalmente en la Sección 5 se discuten las conclusiones del trabajo.

2. PRESENTACIÓN DEL CASO DE ESTUDIO

Se trata de máquinas formadas por módulos de desplazamiento lineal (MDL) que pueden ser ensamblados para obtener diferentes configuraciones, como por ejemplo las mostradas en la Figura 1. Cada MDL consta de un motor eléctrico para mover un cabezal a lo largo del módulo y 5 detectores de posición del cabezal. Además, cada MDL tiene su propio dispositivo de control, compuesto de una tarjeta BeagleBone Black, a la que están conectados los detectores y el motor. De esta forma, el control total de una configuración dada se consigue mediante el sistema de control distribuido (DCS, por sus siglas en inglés) formado por los controladores de cada uno de los módulos que están interconectados mediante una red de comunicaciones. Con esta arquitectura, la reconfiguración mecánica conlleva implícitamente una reconfiguración del *hardware* de control y el cambio entre configuraciones solo requiere añadir o quitar de la red de comunicaciones los dispositivos de control de los módulos correspondientes.

La configuración concreta usada en este estudio se muestra en la Figura 2. Con esta disposición se pretende simular el comportamiento de una máquina que debe realizar una operación dada (estampado, agujereado, etc.) en cuatro puntos diferentes de una pieza. Los puntos para las operaciones son fijos y están situados en las posiciones de los detectores 1 y 5 de los MDL en los ejes X e Y, siendo los puntos concretos (1,1), (5,1), (5,5) y (1,5). El actuador que realiza la operación está en el extremo inferior del MDL del eje Z. Una vez que el actuador está sobre una de estas posiciones, dicho módulo debe bajar hasta la posición 3, donde el actuador se activa automáticamente para realizar la operación. Una vez realizada la operación, la máquina puede pasar al siguiente punto hasta completar las cuatro operaciones. Se considera el estado en que se acaba la última operación como

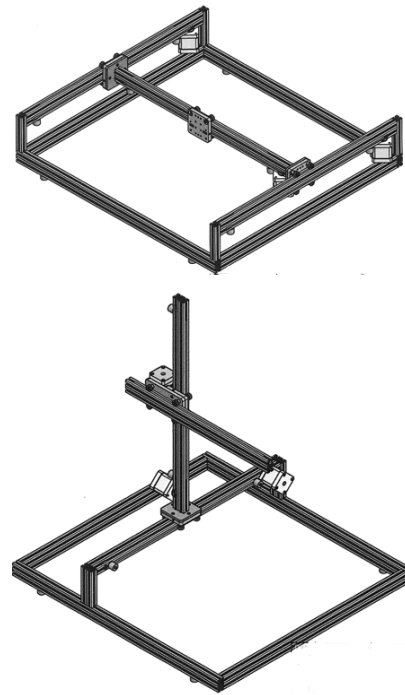


Figura 1: Ejemplos de configuraciones de máquinas mediante el ensamblado de varios MDLs.

estado inicial de la máquina.

El *software* de control resultante que garantiza el comportamiento antes descrito va a depender en gran medida del modelo en base al cual se realiza el mismo. En la siguiente sección se plantean dos modelos en diagramas de Grafcet que modelan el comportamiento de la tarea descrita. A partir de dichos modelos se obtendrán 4 implementaciones del *software* de control que satisface dicho funcionamiento. Las diferencias fundamentales entre las 4 implementaciones estarán en la distribución y en la modularidad de la aplicación de control obtenida en la norma IEC 61499.

Todas las implementaciones se realizarán en el programa open-source 4DIAC [2], que permite desarrollar aplicaciones distribuidas en el estándar IEC 61499. Para todas las soluciones que se describen se va a mantener la configuración de los dispositivos encargados del control, que están conectados entre sí mediante una red Ethernet como se muestra en la Figura 3.

3. Modelos de Grafcet

Para el modelado del comportamiento descrito en la sección anterior se han considerado dos alternativas: la primera consiste en un Grafcet único y la otra considera un modelo jerárquico formado por varios Grafcets. Se ha considerado únicamente el control secuencial de la máquina y se ha excluido

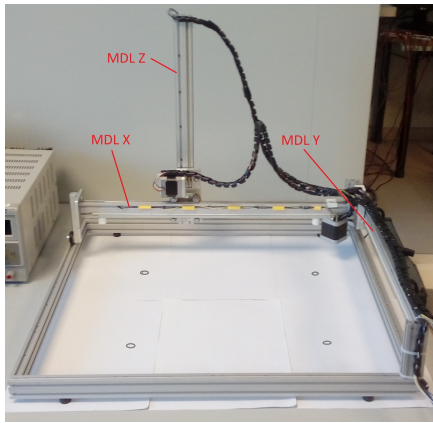


Figura 2: Maqueta real del caso en estudio.

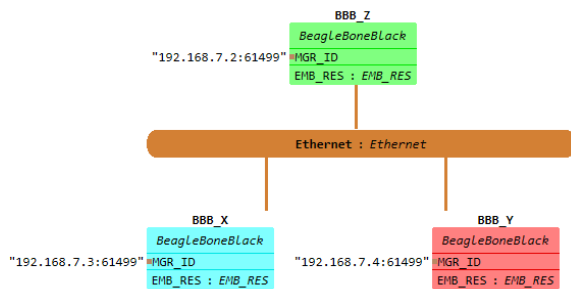


Figura 3: Conexiones entre los dispositivos que forman el DCS donde se implementarán las aplicaciones de control.

el control de los diferentes modos de marcha, paro y emergencia para simplificar el estudio.

El modelado mediante un Grafcet único se muestra en la Figura 4. Este Grafcet integra tanto la secuencia lógica del control como las acciones que cada MDL debe desarrollar. En este caso se considera que el retroceso del MDL Z se realiza automáticamente, por lo que esta fase no se incluirá en la aplicación de automatización. Por otro lado, el modelo jerárquico está formado por un Grafcet principal que define la lógica de funcionamiento, mostrado en la Figura 5 y tres Grafcets que definen el funcionamiento genérico de cada MDL, uno de los cuales se muestra en la Figura 6, donde PX representa la posición actual y TX posición de destino. Para indicar cuándo se ha llegado al destino se ha incluido una variable booleana “XDone” cuya activación dispara las transiciones del Grafcet que define el funcionamiento secuencial. Grafcets similares modelan el comportamiento de los MDLs de los ejes Y y Z.

Una diferencia fundamental entre estos dos modelos es que el modelo jerárquico el Grafcet que define la lógica del funcionamiento se abstrae del funcionamiento concreto de los módulos. Dicho diagrama sólo fija las posiciones a las que se debe

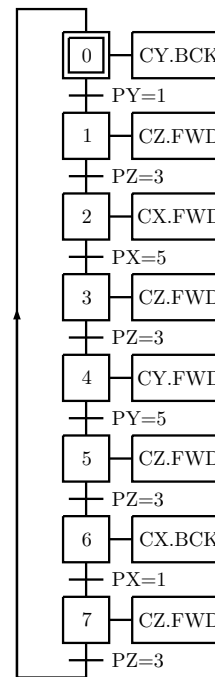


Figura 4: Modelado del sistema mediante Grafcet único.

mover cada módulo en cada momento. Son los diagramas que modelan el comportamiento de cada módulo los que definen la acción concreta (BCK o FWD) en función de la posición actual y de destino. Cabe destacar que para el modelado jerárquico se podría haber usado otras técnicas de interacción entre diagramas, pero se ha elegido la sincronización mediante transiciones por su sencillez para adaptarse a la situación concreta del caso en estudio.

4. Implementaciones del software de control en IEC 61499

A partir de los modelos presentados en la sección anterior se desarrollarán varias implementaciones del *software* de control en la norma IEC 61499. Como cada MDL tiene su propio dispositivo de control, la recogida de datos y las órdenes a los actuadores se dan en el propio dispositivo, propiciando cierto grado de distribución del control, inherente a la estructura del DCS formado por las tres tarjetas BeagleBone Black interconectadas. En lo referente al control lógico secuencial, el algoritmo puede ser implementado de forma centralizada en una de las tarjetas o distribuida entre varias de ellas.

4.1. Implementación 1: algoritmo centralizado y baja modularidad

En primer lugar, podemos considerar un diseño del control secuencial centralizado. Para ello, po-

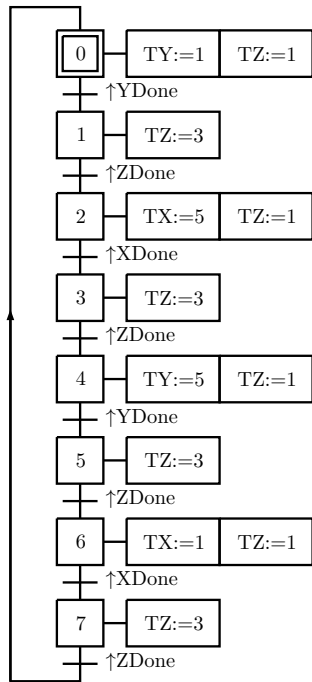


Figura 5: Modelado del sistema mediante jerarquía de Grafcets. Grafcet de control secuencial.

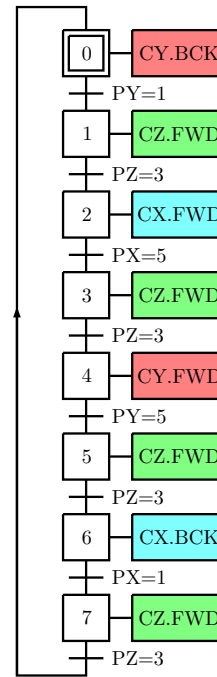


Figura 7: Modelo de Grafcet único mostrado en la Figura 4 indicando la distribución de las acciones en los MDLs.

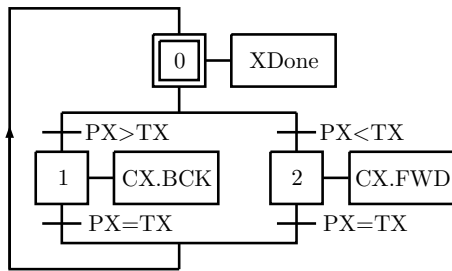


Figura 6: Modelado del sistema mediante jerarquía de Grafcets. Grafcet que modela el funcionamiento del MDL del eje X.

demos usar directamente el modelo de Grafcet presentado en la Figura 4, el cual se muestra nuevamente en la Figura 7 resaltando las acciones que afectan a cada dispositivo con su respectivo color de acuerdo con la Figura 3.

Este diagrama de Grafcet se ha implementado en la norma IEC 61499 mediante el *Execution Control Chart* (ECC) de un bloque de funciones básico según los patrones de traducción de los elementos de Grafcet a IEC 61499 propuestos en [8], Figura 8. El bloque de funciones resultante junto con la aplicación que implementa el control se presentan en la Figura 9.

El bloque *CentralizedCtrl* que implementa el control descrito en el Grafcet propuesto consta de tres entradas de datos pertenecientes a las posiciones medidas en cada eje y de tres salidas corres-

pondientes a la acción de control para cada eje. La aplicación muestra la distribución de los elementos, que viene indicada por el color de los bloques de funciones, que se corresponde con el color del dispositivo que lo ejecuta según se representa en la Figura 3. De esta forma las subaplicaciones *Read* y los bloques que implementan la salida booleana o de PWM de la tarjeta de control serán ejecutados por la tarjeta sobre la que deben medir y actuar. Los bloques mencionados se deben mapear en estos dispositivos ya que las entradas y salidas están justamente en estos dispositivos. Sin embargo, el bloque *CentralizedCtrl* podría ejecutarse en cualquiera de los tres dispositivos, en este caso se ha asignado al dispositivo que controla el eje Z.

Esta implementación centralizada de la aplicación conlleva que todos los datos de posición sean transmitidos a la tarjeta del eje Z y que esta envíe las acciones de control oportunas al resto de tarjetas. A su vez, si se desean realizar modificaciones en la aplicación de control solo se debe modificar el bloque *CentralizedCtrl* que implementa el modelo de Grafcet.

4.2. Implementación 2: algoritmo distribuido y baja modularidad

En el apartado anterior, si bien existía cierta distribución en distintos dispositivos en cuanto a la recepción y envío de datos, el algoritmo de control se encontraban únicamente en un dispositivo,

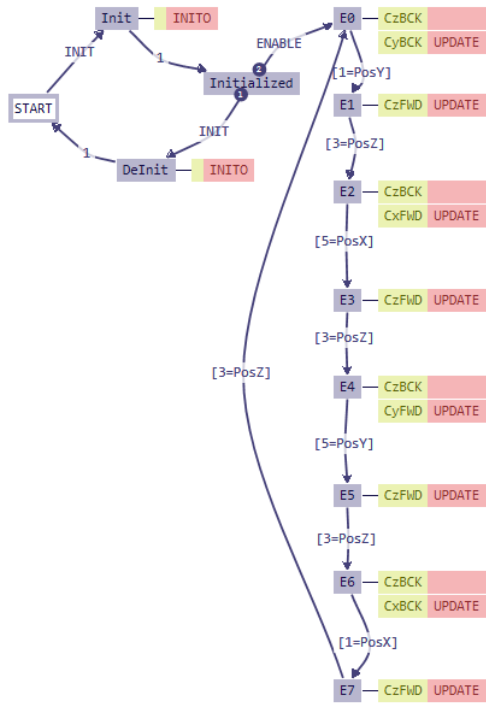


Figura 8: ECC que implementa el Grafcet de control secuencial de la Figura 4.

el que controla el eje Z. Gracias a la flexibilidad en la distribución del código que aporta el estándar IEC 61499, esto puede no ser así, es decir, distintos dispositivos pueden encargarse simultáneamente de ejecutar una aplicación de control.

En este caso mantendremos como base de referencia el modelo de Grafcet mostrado en la Figura 4. Manteniendo la misma estructura, en la Figura 10 se presentan tres diagramas de Grafcet, cada uno coloreado acorde al dispositivo en el que se van a implementar. Todos presentan las mismas etapas y transiciones pero cada uno de ellos sólo incluyen las acciones de control correspondientes al dispositivo donde será implementado. La funcionalidad de la aplicación se mantiene ya que 4DIAC permite que los bloques que implementan estos modelos de Grafcet se puedan ejecutar simultáneamente.

Por lo tanto, una aproximación sistemática para la distribución del control consiste en esta división en varios diagramas de Grafcet para luego implementar cada uno de ellos en bloques de funciones independientes.

Se han desarrollado tres bloques de funciones llamados *Ctrl_X*, *Ctrl_Y* y *Ctrl_Z*, cada uno conteniendo sus respectivas acciones de control. Estos bloques se han introducido en la aplicación que se muestra en la Figura 11 y se han mapeado cada uno en su respectivo dispositivo.

La aplicación resultante implementa los algorit-

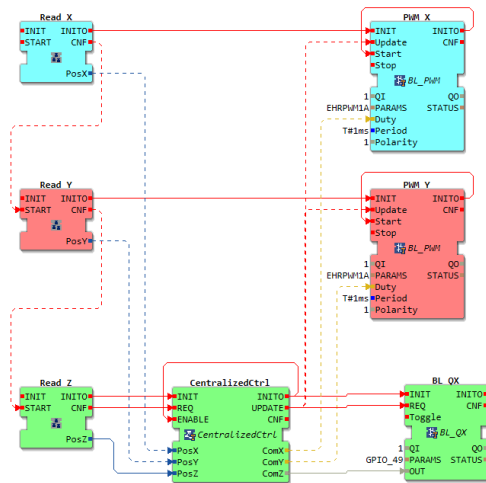


Figura 9: Implementación en 4DIAC de la aplicación de control centralizada.

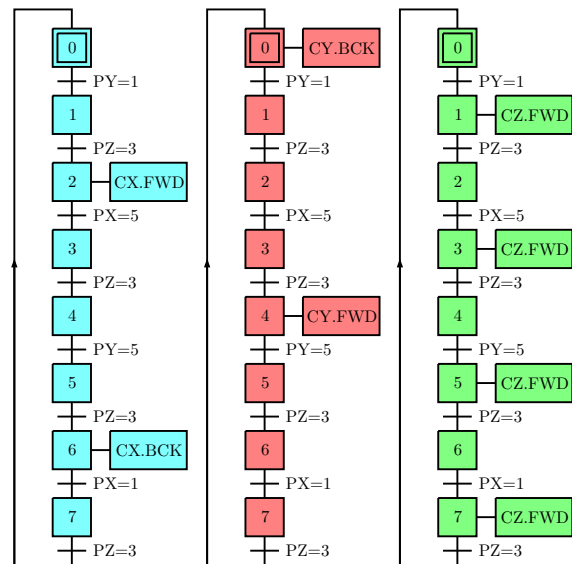


Figura 10: Distribución en tres Grafcets del Grafcet original presentado en la Figura 7.

mos de control de forma distribuida, siendo estos ejecutados en todos los dispositivos simultáneamente. Sobre la transmisión de datos, esta distribución implica que los valores de posición de los ejes ahora tienen que ser transmitidos a todos los dispositivos, a diferencia del caso anterior, en el que se transmitían todos a un solo dispositivo. Sin embargo, las acciones de control resultantes se aplican directamente dentro del mismo dispositivo, no requiriendo así transmisión de las mismas. En el caso anterior, las acciones de control debían ser transmitidas a los dispositivos pertinentes.

Por otra parte, en el caso de modificarse la aplicación de control se deberían modificar los tres bloques de funciones, a diferencia del caso anterior, en el que solo se modificaría un bloque. Esto,

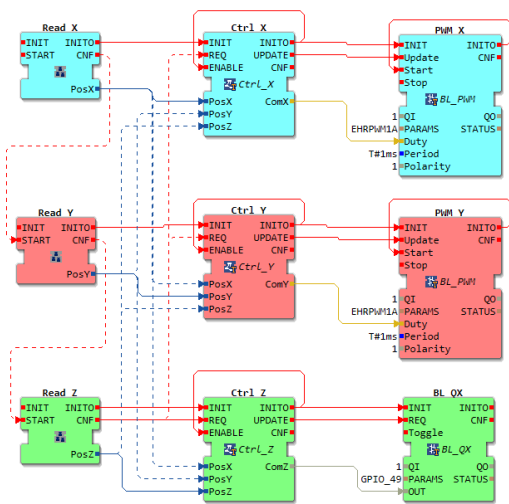


Figura 11: Implementación en 4DIAC de la aplicación de control distribuida.

desde el punto de vista de la flexibilidad y reconfigurabilidad del *software* de control, supone un claro inconveniente.

4.3. Implementación 3: algoritmo centralizado y modular

Las implementaciones 1 y 2 se realizaron a partir de un modelo de Grafset único que contiene tanto el control lógico secuencial como las acciones concretas a desarrollar por cada eje. Este enfoque es práctico para la aplicación concreta ya que el funcionamiento de cada MDL es sencillo, sin embargo, para otras máquinas o módulos las acciones pueden ser más complejas.

Para la implementación 3 consideraremos el modelo jerárquico presentado en la sección 3 en el que se separa el funcionamiento genérico de un MDL del control secuencial que coordina la operación de la máquina. El Grafset que describe el funcionamiento de un MDL (Figura 6) se ha implementado mediante un bloque de funciones básico y, junto con los bloques de funciones encargados de obtener la posición del eje y actuar sobre el mismo (READ y PWM) se han encapsulado en una subaplicación. Esta subaplicación, que tiene como entrada la posición de destino a la que debe ir el eje y como salida la variable booleana Done, puede ser considerada como un *driver* del MDL.

Por otro lado, el Grafset que describe el control secuencial (Figura 5) se implementará de forma centralizada en un único dispositivo, aunque evidentemente, las acciones serán desarrollada por los dispositivos de control de cada módulo. Esto se ha destacado en la Figura 12 coloreando las acciones que afectan a cada uno de los dispositivos con su color respectivo.

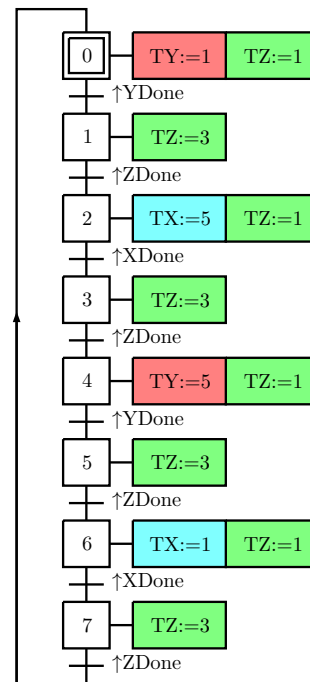


Figura 12: Modelo de Grafset jerárquico (Figura 5) indicando la distribución de las acciones.

La aplicación en 4DIAC que implementa este control se presenta en la Figura 13, en la que cada subaplicación de los MDL se ha mapeado en su respectivo dispositivo. Sin embargo, como en el primer caso, el bloque *Ctrl_Log_Centr* que implementa el modelo de Grafset de la Figura 12 puede ser mapeado en cualquier dispositivo, y en este caso se ha asignado a la BeagleBone Black que controla el MDL del eje Z.

Esta implementación modular favorece el reemplazo de los módulos ya que la aplicación a desarrollar no cambia, con lo cual, si se aporta el *driver* correspondiente al nuevo módulo los cambios en la aplicación son mínimos.

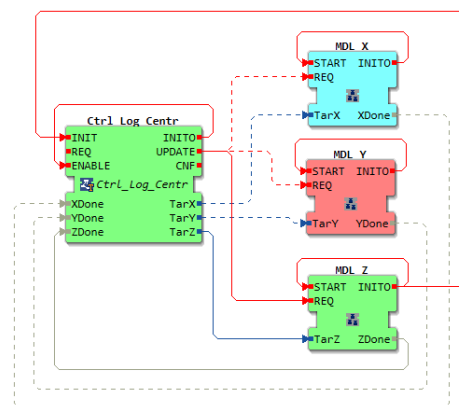


Figura 13: Implementación de la aplicación modular y con algoritmos centralizados.

4.4. Implementación 4: algoritmo distribuido y modular

La aplicación del apartado anterior presenta el algoritmo de control secuencial centralizado en un solo dispositivo. Sin embargo, este puede ser distribuidos entre los diferentes dispositivos encargados del control de los módulos. Para ello partimos del diagrama de Grafset propuesto en la Figura 12. Tal y como se ha visto en la Sección 4.2, este diagrama de Grafset se puede dividir en tres diagramas con la misma estructura (etapa y transiciones), pero cada uno de ellos solo con las acciones que se deben desarrollar en cada dispositivo. En la Figura 14 se recogen los tres diagramas de Grafset que permiten obtener una aplicación distribuida.

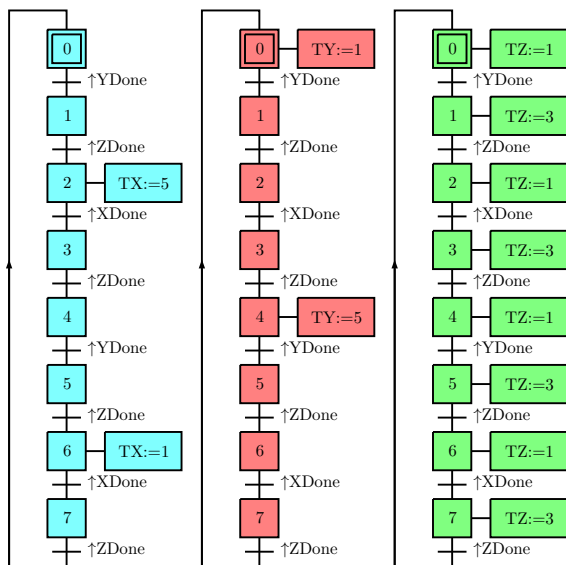


Figura 14: Distribución en tres Grafsets del Grafset modular centralizado de la Figura 12.

Cada uno de estos diagramas de Grafset se ha implementado en un bloque de funciones básico, llamados *Ctrl_Log_X*, *Ctrl_Log_Y* y *Ctrl_Log_Z*, y se han mapeado en su correspondiente dispositivo como se muestra en la aplicación de la Figura 15.

Respecto al control de cada uno de los MBL, se mantienen las subaplicaciones que implementan el control como se modela en la Figura 6.

En la aplicación resultante los datos sobre el estado de los ejes, los booleanos “Done”, se transmiten entre todos los dispositivos, sin embargo, la posición de destino a transmitir a cada eje queda dentro del propio dispositivo. Esto mismo sucedía en el segundo caso mostrado. Sin embargo, a diferencia de aquella implementación, si en esta se desea modificar la aplicación, por ejemplo para cambiar las posiciones de un eje en las que la máquina tiene que hacer las operaciones, solo

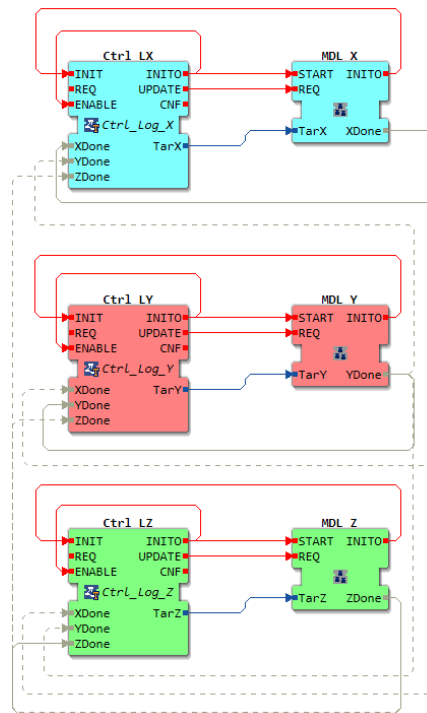


Figura 15: Implementación de la aplicación modular distribuida.

es necesario modificar las posiciones de destino de dicho eje, quedando el resto de bloques intactos.

Además, como en el caso modular centralizado, cada módulo puede ser reemplazado o modificado, modificando sus respectivos *drivers*, sin tener que alterar los algoritmos de control presentados en la Figura 14.

5. CONCLUSIÓN

En este trabajo se presentan diferentes implementaciones del *software* de control en la norma IEC 61499 a partir de una descripción funcional en modelos de Grafset. Se considera un caso de estudio que consiste en una máquina compuesta por tres módulos de desplazamiento lineal controlados cada uno de ellos por tarjetas independientes.

Dos modelos de Grafset han sido propuestos, uno de ellos con una concepción más centralizada del diseño de sistemas secuenciales y otro centrado en una concepción más modular. Estos modelos han servido de base para la implementación de las aplicaciones en la norma IEC 61499, proponiéndose dos implementaciones por cada modelo, y centrandolo la atención en la distribución de los algoritmos de control que permite esta norma.

El estudio de las cuatro implementaciones permite hacer las siguientes observaciones. En cuanto a la distribución de las funcionalidades de la máqui-

na, en el caso del modelo mediante Grafcet único, la función de los dispositivos de control de cada módulo se limita a la lectura de sensores y activación de salidas. Por otro lado, el modelo jerárquico permite, además de las operaciones relacionadas con sensores y actuadores, definir en el dispositivo de control de cada módulo su comportamiento genérico, que consiste en moverse hacia las posiciones indicadas. En ambos modelos, el código que define el comportamiento secuencial del la máquina puede ser o no distribuido entre los dispositivos de control de cada módulo. Esta distribución se consigue replicando la estructura del modelo de Grafcet en cada dispositivo, pero incluyendo únicamente las acciones correspondientes a cada uno.

Podemos concluir que el modelo de Grafcet influye en el *software* de control obtenido mediante la norma IEC 61499. En particular, el caso presentado se beneficia de las ventajas del diseño basado en componentes a través de la implementación del modelo de Grafcet jerárquico, ya que los módulos de desplazamiento lineal se pueden identificar claramente como componentes. Sin embargo, al tratarse de una aplicación donde los diferentes módulos deben interactuar en múltiples ocasiones durante un ciclo de operación, la distribución de los algoritmos implica un intercambio de datos constante entre los módulos de control. Estas características de los sistemas y de las aplicaciones deben ser tenidas en cuenta en el diseño e implementación de los sistemas de automatización.

Agradecimientos

Este trabajo ha sido financiado por la Universitat Jaume I con nº de proyecto 18I411-UJI-B2018-39 y por la CEICE con nº de beca ACIF/2018/244.

English summary

GRAF CET MODELS AND DISTRIBUTED APPLICATIONS IN IEC 61499 STANDARD. A CASE STUDY.

Abstract

In this work a case study is presented in which the behaviour of a modular machine is modeled with different Grafcet models and the control software is implemented according to the standard IEC 61499 that enhances the design of distributed applications. The proposed models take into account different degrees of modularity of the components and, based on these models, different implementations are proposed, seeing how the models influence the distribution of the control algorithms.

Keywords: Grafcet, IEC 61499, DCS

Referencias

- [1] René David and Hassane Alla. *Discrete, continuous, and hybrid Petri nets*, volume 1. Springer, 2005.
- [2] Eclipse 4diac. Eclipse 4diac - the open source environment for distributed industrial automation and control systems, 2020.
- [3] IEC. IEC 61499: Function blocks for industrial-process measurement and control systems, 2000.
- [4] IEC. IEC 60848: GRAFCET specification language for sequential function charts, 2002.
- [5] IEC. IEC 61131 - programmable controllers, part 3: Programming languages, 2013.
- [6] Italia Jiménez, Ernesto López, and Antonio Ramírez. Synthesis of ladder diagrams from Petri nets controller models. In *Proceeding of the 2001 IEEE International Symposium on Intelligent Control (ISIC'01)*, pages 225–230. IEEE, 2001.
- [7] Robert Julius, Max Schürenberg, Frank Schumacher, and Alexander Fay. Transformation of GRAFCET to PLC code including hierarchical structures. *Control Engineering Practice*, 64:173–194, 2017.
- [8] Oscar Miguel-Escrig, Julio-Ariel Romero-Pérez, Bianca Wiesmayr, and Alois Zoitl. Distributed implementation of Grafquets through IEC 61499. In *2020 25th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 2020.
- [9] Frank Schumacher and Alexander Fay. Formal representation of GRAFCET to automatically generate control code. *Control Engineering Practice*, 33:84–93, 2014.
- [10] Alois Zoitl and Robert W. Lewis. *Modelling control systems using IEC 61499*, volume 95 of *IET Control engineering series*. IET, London, 2. ed. edition, 2014.



© 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).