

APLICACIÓN DE IMRP EN AUTOMATISMOS BASADOS EN CONTROLADOR LÓGICO PROGRAMABLE

Acosta-Cano-de-los-Ríos José E. Rodríguez Hermosillo Silvia
Tecnológico Nacional de México/Instituto Tecnológico de Chihuahua
División de Posgrado e Investigación
Ave. Tecnológico No. 2909
614 201 2000 ext. 2112
jacosta@itchihuahua.edu.mx

RESUMEN.

Se presenta una solución al problema de reusabilidad de código en automatismos basados en controlador lógico programable. El paradigma orientado a objeto (POO) permite el desarrollo de sistemas modulares y flexibles, características que han sido aplicadas durante décadas en sistemas informáticos con lenguajes de alto nivel. Recientemente ha sido de interés la aplicación del POO en sistemas basados en PLC para apoyar el diseño e implementación del sistema, bajo el esquema de implementación dirigida por modelo, es decir, a partir del modelo (OO) se genera el código del sistema (semi o automáticamente). En este tipo de esfuerzos la reusabilidad no se enfoca principalmente al código sino a la reusabilidad del modelo. Otra forma de reusabilidad puede ser el uso del mismo código en diferentes aplicaciones particulares. En este sentido en el presente trabajo se plantea el uso del modelo de particularidades para ser interpretado por un sistema genérico, en donde el mismo código (genérico) es reutilizado en diferentes aplicaciones, con base en el modelo de particularidades, bajo el esquema de operación dirigida por modelo. Para tal efecto, la técnica iMRP hasta ahora utilizada en el desarrollo de sistemas a nivel célula, es analizada y aplicada a modelar particularidades a nivel lógica de operación entre sensores y actuadores. iMRP es de utilidad en este tipo de modelado, sin embargo, el nivel de madurez de las plataformas para desarrollo de automatismos basados en PLC no es suficiente para la implementación y operación del sistema genérico, por lo que su aplicación en el presente proyecto requiere de un sistema coordinador operando en forma integrada con una PC.

ABSTRACT.

A solution to code reusability problem of programable logic controller is proposed. Object oriented paradigm (POO) supports development of modular and flexible systems. These characteristics have been achieved for decades in software system development using object oriented high level languages. Recently, the development of this type of systems based on POO has been of interest in the PLC system arena. The main effort in this direction has been using the POO toward model driven implementation concept, where based on OO models the system code is generated (semi or automatically). This type of solutions are focused mainly on model reusability rather than code reusability. It implies using

the same code (a generic code), to apply the system to generate several particular systems. Toward that end this article proposes using the concept of particularities model based on the iMRP modeling technique for PLC based systems. A particularities model is to be interpreted by the generic system to instantiate a particular application. iMRP technique has been applied successfully on several production cells. In this article an application of iMRP is proposed to PLC code reusability, abstracting system elements at sensor and actuators level. Since PLC development platforms do not fully support object oriented programming, model driven operation requires PC support for the execution of the generic system.

Keywords: OOP, PLC, Operación dirigida por modelo, iMRP, Flexibilidad.

1. INTRODUCCIÓN

La reusabilidad de hardware ha alcanzado bastante solidez no sólo en el área de sistemas basados en PC sino también en plataformas basadas en PLC. Por otra parte, el tema de reusabilidad de software en sistemas basados en PLC se encuentra con avances modestos debido al nivel de maduración de los lenguajes de programación y sus respectivos entornos de desarrollo. Una de las principales carencias en esta plataforma (PLC) son las características asociadas al paradigma orientado a objeto (POO), [1], que soportan el reuso de código;. Los PLC son programados bajo el estándar IEC 61131-3 [2], el estándar apoya con características de POO en la programación de PLC's, como es el caso de bloques funcionales. Existen lenguajes para el desarrollo orientado a objetos de este tipo de sistemas, como son UML, SysML y XML. Otras técnicas se orientan a modelar el comportamiento de los sistemas como son las redes de Petri, GRAFCET, SFC, iMRP [3][4][5][6] entre otras. Sin embargo, los esfuerzos reportados en la literatura se orientan a generar código de manera automática a partir del modelo del sistema, concepto identificado como implementación dirigida por modelo, [7]. En este sentido el modelo (generalmente de naturaleza gráfica) es reutilizable para diferentes controladores.

Acotando la reusabilidad en el programa mismo, sería que el mismo programa fuera de aplicación en sistemas que presenten diferencias pero corresponden a una misma naturaleza de sistema, esto es, ajustándose al concepto de familia, como lo es la línea de productos [8] así el programa sería reutilizable en una familia de sistemas, dando como resultado un sistema de aplicación genérica.

El objetivo del presente trabajo de investigación es plantear una solución al problema de la aplicación de un sistema genérico (basado en PLC) mediante la adaptación del (mismo) código a diferentes casos particulares. El programa (sistema) debe ser adaptado a nuevas aplicaciones sin requerir cirugía mayor en su código. Para tal efecto se plantea la particularización del sistema genérico utilizando el concepto de operación dirigida por modelo.

La escritura del presente artículo está organizado de la siguiente manera. En la sección 2 se plantea el concepto de modelado de particularidades bajo la técnica de iMRP en células de producción. En la sección 3 se describe la propuesta conceptual de solución y el modelado de la lógica del sistema utilizando la técnica de diagramas de estados finitos. En la sección 4 se aplica la propuesta de solución bajo el concepto de operación dirigida por modelo. Finalmente se describen las conclusiones obtenidas en la realización del proyecto.

2. MARCO TEÓRICO

2.1. Técnica de modelado iMRP.

En el esquema de referencia ArquiTAM [9]), se plantea que una aplicación informática de uso genérico (conocida en el mundo anglosajón como framework) permita instanciar un sistema de aplicación particular (con base en el paradigma orientado a objetos) donde el sistema resultante recibe el nombre de modelo particular o sistema de aplicación particular. Generalmente hablando el modelado es utilizado para simplificar el entendimiento, diseño y operación de sistemas, mediante la abstracción de la información relevante. Como se mencionó en la sección anterior existe una amplia variedad de técnicas de modelado. La técnica iMRP (acrónimo de integrador modular de recursos de producción) planteada en [10] se orienta a obtener un modelo interpretable por una aplicación informática bajo el concepto de operación dirigida por modelo (no implementación dirigida por modelo). Esta técnica parte del concepto de abstraer (modelar) únicamente las características específicas del sistema en particular sin tomar en cuenta las características comunes en el tipo de sistema a modelar. Las características comunes ya se encuentran implementadas (en código) en una aplicación informática genérica en forma de marco de trabajo (conocida en el mundo anglosajón como framework). La aplicación informática de uso genérico (framework) se personaliza o particulariza mediante la instanciación de los diferentes objetos del sistema particular mediante la interpretación del modelo iMRP del sistema particular así mismo ejecuta las operaciones con base en la

lógica establecida en el modelo iMRP. Es importante destacar que en el modelo iMRP además de identificar los objetos que forman al sistema particular, identifica las clases (implementación en código) de los mismos.

iMRP es una técnica gráfica que consiste en la abstracción de particularidades de una célula o línea de producción, en donde todas estas particularidades se encuentran abstraídas en forma gráfica ya sea en un nodo o en un arco, en donde cada nodo tiene atributos como tipo (clase), nombre, nivel y el arco presenta atributos como, GrafoID, cantidad, variables dependientes del concepto (var1, Var2 y var3). El modelo desarrollado en iMRP presenta de una manera sencilla las características específicas del sistema y de una manera fácil de interpretar por una aplicación informática.

El modelo de particularidades abstrae las características específicas del sistema de manufactura tales como equipos de producción, operaciones a realizar y condiciones de precedencia. El modelo contiene cuatro sub grafos (Producto – P-, Recurso Tiempo –RT-, Célula –C- y Operaciones –O-), enlazados a través de los nodos del grafo RT. El grafo P describe la estructura del producto conocida por el acrónimo anglosajón BOM (lista de materiales). El nodo de nivel inferior representa la materia prima, el nodo de nivel superior representa el producto terminado El grafo O incluye los nodos correspondientes a los cuatro tipos de operaciones definidas en la arquitectura de referencia [9] [10], implementadas en el marco de trabajo (manipulación, transporte, procesamiento y alimentación/retiro).

A manera de ilustración del modelado iMRP a continuación se presenta una síntesis del ejemplo planteado en (tesis). El ejemplo consiste en la abstracción iMRP de una célula formada por un robot (CRS A465) y un torno (Dyna DM3300). El modelo iMRP sería utilizado (interpretado) por una aplicación informática lanzador de órdenes para coordinar la operación de cada uno de los dos equipos. La operación de la célula consiste en colocar mediante el robot una pieza en el torno para su procesamiento luego de ser procesada retirarla del torno para colocarla en una caja con piezas terminadas. En la Fig. 1 se muestra el modelo iMRP de la célula.

El sub-grafo Célula contiene los nodos que representan a los equipos de producción (robot y torno). Los nodos del sub-grafo RT (en combinación con los nodos del sub-grafo P) apoyan en la descripción de las condiciones de precedencia para la realización de las operaciones.

A cada uno de los nodos del sub-grafo RT se asocian tres nodos, uno por cada uno de los tres sub-grafos restantes para describir la pieza a operar, la operación a realizar y el equipo a utilizar. Por ejemplo, mediante los nodos asociados al nodo RT1 se describe que la pieza B será operada por el equipo R (robot) en una operación de manipulación (M). La secuencia de operación es establecida, en parte por la estructura padre-hijo del grafo P, de manera tal que el padre no puede ser operado sin haber operado antes la totalidad de sus hijos (componentes). Además, en el caso que una pieza requiera más de una

operación la secuencia de operación es establecida por la estructura padre-hijo en el sub-grafo RT. Por ejemplo, la pieza A (RT2), requiere la operación de procesamiento (nodo P) a realizar por el torno (nodo T) y posteriormente (nodo hijo RT3) será operada mediante una operación de manipulación (nodo M) por el robot (nodo R) para ser retirada del torno.

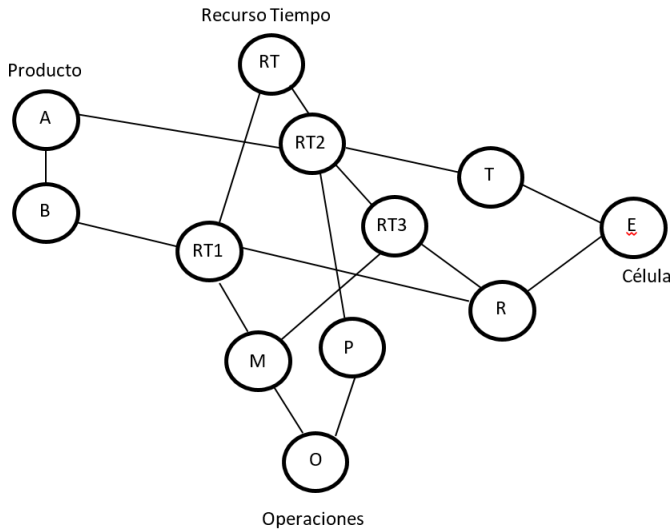


Fig. 1. Modelo de particularidades iMRP de la Célula robotizada (adaptado de [11]).

Así el modelo de particularidades contiene la estructura y lógica de operación del sistema específico, información a ser interpretada en tiempo de ejecución por el sistema genérico o lanzador de órdenes (operación dirigida por modelo).

Es importante destacar que el primer paso en la operación del lanzador de órdenes consiste en instanciar los objetos correspondientes al sistema en particular, a partir de las clases implementadas en el marco de trabajo y la información contenida en el modelo de particularidades (tipo – clase); en éste se indica la clase que implementa al tipo de objeto a instanciar. Esta información se encuentra en los atributos de cada nodo como se muestra en la Fig. 2. Por ejemplo, la clase a utilizar para instanciar el objeto representante del torno está indicada como C:/Framework/CProcesador1, en el parámetro Tipo de los atributos del nodo T.

Así mismo, los programas a ejecutar por los equipos están definidos en el atributo VAR1 del arco que asocia a los nodos del grafo RT con los nodos del grafo O. Por ejemplo, la pieza B (nodo B) asociada al nodo RT1, en la Fig. 3, es operada por el robot (nodo R) en una operación de manipulación. En el arco que asocia a los nodos RT1 y M, Fig. 3, el atributo VAR1 contiene el nombre del programa a ejecutar por el robot en esta operación. Cada una de las operaciones de manipulación a realizar por el robot (carga o descarga del torno) requiere un programa diferente, como se indica en los atributos VAR1 de los dos arcos que asocian a los nodos RT1 y RT3 con el nodo M (C:/Robot/Alim_Torno y C:/Robot/T_Banda

respectivamente). De la misma manera se define en el atributo VAR1 del arco respectivo el programa a utilizar por el torno en la operación de procesamiento (C:/CNC/PiezaA).\

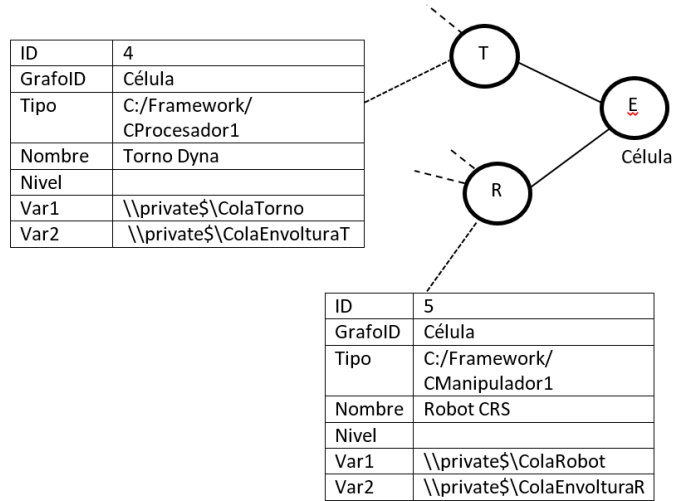


Fig. 2. Atributos nodo en iMRP.

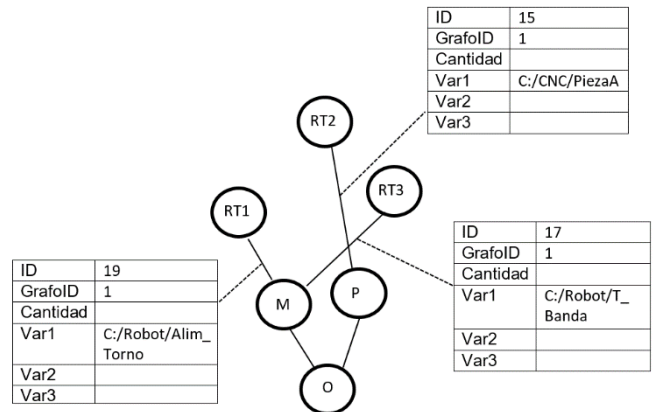


Fig. 3. Atributos arco de iMRP.

2.2. Diseño tradicional basado en máquinas de estados finitos.

Diagramas de estados finitos es una herramienta común para el desarrollo de sistemas informáticos, facilita el desarrollo de programas y la documentación de los mismos, situación extensible al caso de programas basados en PLC. Existen herramientas que permiten generar código a partir de este tipo de diagramas, para programas basados en PC, (ejemplo: SineLABORE), es decir, en esta técnica que también es de utilidad en el desarrollo de sistemas basados en PLC, a partir del modelo se desarrolla el código del programa. A continuación se describe el diagrama de estados finitos para un sistema de llenado. Con la finalidad de ver el efecto en el

programa debido a cambios en el sistema, se presenta un sistema ligeramente modificado y su diagrama de estados correspondiente.

El sistema a controlar consiste en una tolva y una banda transportadora para el llenado de recipientes como se muestra en la Fig. 4. El sistema cuenta con dos sensores (presencia y llenado) y un actuador para abrir/cerrar la válvula de la tolva. El diagrama de estados finitos se muestra en la Fig. 5.

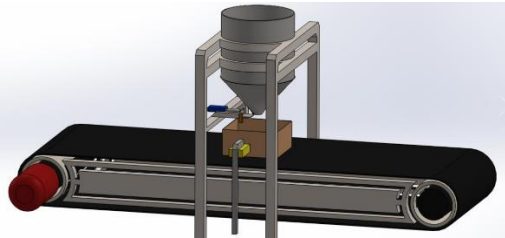


Fig. 4. Sistema de envasado de una tolva.

Una modificación leve al sistema consiste en colocar una segunda tolva, donde la caja requiere cargar material de Tolva1 y de Tolva2, Fig. 6. Con la intención de conservar de una manera sencilla el modelado, se considera que sólo una caja se encontrará presente en la banda transportadora a la vez. El diagrama de estados finitos para el sistema modificado se muestra en la Fig. 7. Es importante destacar el cambio considerable entre los modelos (diagramas) Figs. 5 y 7, lo cual implicaría un esfuerzo considerable para el rediseño del código en el PLC.

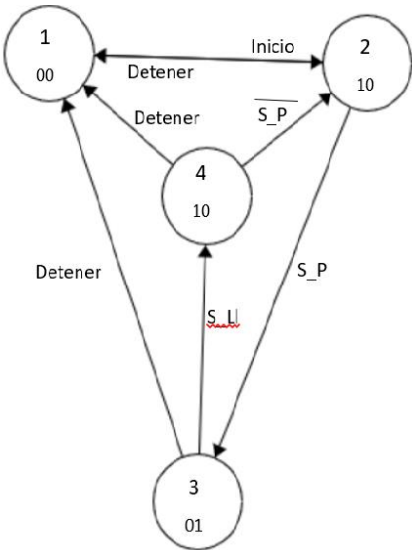


Fig. 5. Diagrama de estados finitos para el sistema de envasado de una tolva.

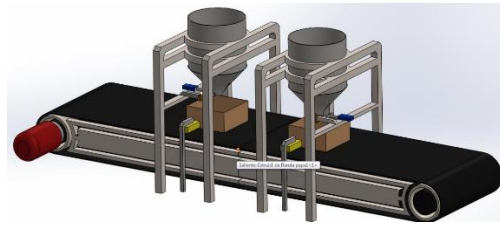


Fig. 6. Sistema de envasado modificado con dos tolvas.

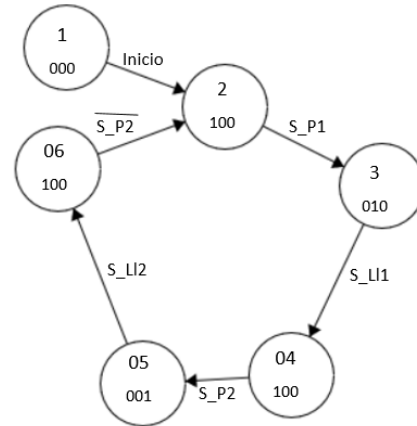


Fig. 7. Diagrama de estados finitos del sistema de envasado de dos tolvas.

3. PROPUESTA DE SOLUCIÓN.

3.1. Instanciación de objetos en PLC

El concepto de clase no es soportado en los entornos de desarrollo para PLC, como lo es el caso del soporte en lenguajes de alto nivel en el área de PC. En [12] se plantea una plataforma apoyada en computadora personal para la implementación de clases y creación de objetos, dirigida a facilitar la reusabilidad de código en sistemas basados en PLC. El planteamiento consiste en implementar las clases en lenguaje de alto nivel (ejemplo: vb.Net), así como también la instanciación de objetos. Posteriormente cargar los objetos instanciados en el PLC.

La idea básica consiste en que la clase implementada en el lenguaje de alto nivel (Vb NET) ofrezca la funcionalidad para generar el texto correspondiente a una instancia de objeto en el archivo del programa del PLC (.L5K). La aplicación en VB agregará texto al archivo .L5K que representa una lógica en el PLC. La clase tiene los atributos (propiedades) que corresponden a las secciones TAG y Program del archivo .L5K, Fig. 8.

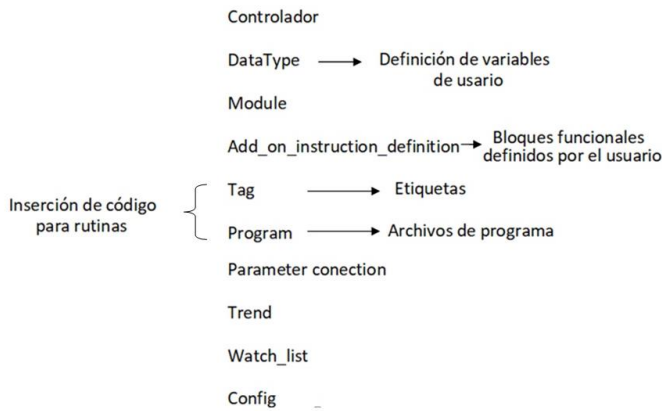


Fig. 8. Programa en PLC Rockwell en archivo .L5K.

En cada clase se encuentran etiquetas (TAGS) correspondientes al acceso de la funcionalidad implementada en la clase, (ejemplo: TAGs de paro, arranque y salida de motor), junto con el texto correspondiente a la lógica de escalera de la clase, (ejemplo: clase arrancador, banda, etc.). Los valores de las variables TAG son particulares de cada instancia de objeto y de acuerdo a la conexión de los sensores y actuadores en el PLC (entradas/salidas).

En el área de ingeniería de software se encuentran planteamientos dirigidos a apoyar el desarrollo de sistemas en alguna de sus etapas. Algunos de estos planteamientos parten de la idea de generar código a partir del modelo (ejemplo UML), esto es, implementación dirigida por modelo. En este sentido se encuentra la técnica conocida como MDA (acrónimo en inglés de Arquitectura dirigida por modelo); otras soluciones como la conocida como línea de productos, son propuestas a partir de un sistema base codificado para generar una especialización del mismo para adaptarlo a una situación particular, [8]. En el esquema de referencia ArchiTAM [9] se plantea hacer uso del concepto de operación dirigida por modelo, esto es, a partir de un sistema informático codificado, interpretar en tiempo de ejecución el modelo de particularidades del sistema específico para generar las instancias correspondientes (instanciación en lugar de especialización) y realizar la operación del mismo.

El concepto clase-instanciación permite reutilizar módulos (clases) de código en el sistema informático que son personalizados en tiempo de ejecución, tanto en sus propiedades (TAGs) como en la cantidad de los mismos. De tal manera es posible contar con clases implementadas en un sistema (genérico) que coordine la creación de objetos y la ejecución del código; tales sistemas se consideran sistemas genéricos, conocidos en el mundo anglosajón como frameworks; sistemas que facilitan el desarrollo de nuevos automatismos

4. OPERACIÓN BASADA EN IMRP –OPERACIÓN DIRIGIDA POR MODELO-

En la operación dirigida por modelo el modelo de un sistema particular es interpretado por un sistema de aplicación general (no particularizado), para la instanciación de elementos (generar sistema particular) y coordinación de la operación de los mismos, [9] [11].

La técnica de modelado iMRP permite abstraer sólo las particularidades de un sistema, tomando en cuenta que en el sistema genérico se encuentran implementadas las características comunes a todos los sistemas de la misma naturaleza; de esta manera se logra reducir la complejidad del modelo y facilitar su interpretación por el sistema informático (genérico). En la Fig. 9, se muestra el modelo iMRP para la operación del sistema con una tolva.

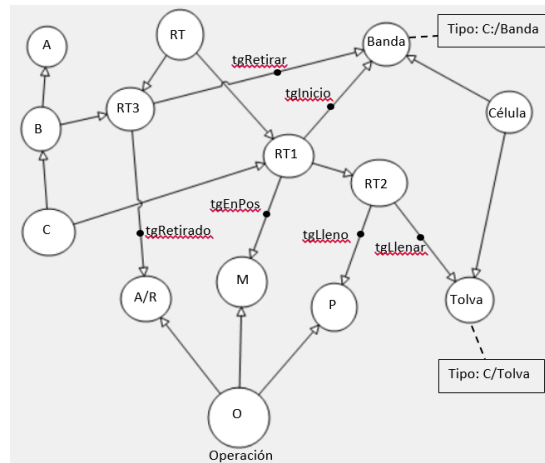


Fig. 9. Modelo iMRP de sistema de envasado.

El sistema genérico (coordinador) interpreta en tiempo de ejecución el modelo iMRP de particularidades del sistema para instanciar (generar el código) de los objetos actuadores (equipos) en el archivo .L5K, que forma el programa a cargar en el PLC, [12]. En el atributo Tipo de los nodos equipo se identifica la clase (librería de clases) en donde se encuentra implementado el código del PLC a insertar en el archivo .L5K. Así mismo el código de la clase se incorpora al sistema genérico (programa en vb NET) mediante incorporación dinámica de código. Una vez cargado el programa con los objetos (clases definidas en el atributo Tipo del nodo equipo correspondiente), el sistema genérico mediante las etiquetas correspondientes ejecuta las funciones de los objetos, de acuerdo a la lógica establecida en el modelo iMRP. Es importante mencionar que el sistema genérico está operando en una PC en lenguaje de alto nivel (en este caso vb NET).

La lista de fases en la operación del sistema se describe en el grafo con los nodos A, B y C. La fase C es la primera en realizarse utilizando los recursos tiempo RT1 y RT2. El recurso tiempo RT1 está formado por el equipo banda (nodo Banda) y la operación de manipulación (nodo M). El atributo del nodo

Banda especifica la clase que contiene el código a insertar en el archivo programa (PLC), correspondiente al manejo de la banda. La ejecución de tal código inicia con asignar (por parte del coordinador) el valor verdadero a la etiqueta tgRetirar, como se indica en el grafo; así la banda comienza a moverse hasta cuando el sensor de proximidad indica la posición de la caja debajo de la tolva. En esta posición el código de la clase Banda activa (verdadero) a la etiqueta tgEnPosición. De esta manera el coordinador del sistema identifica que el recurso tiempo RT1 de la fase C ha sido concluido. A continuación se realiza el recurso tiempo RT2, formado por el equipo Tolva (nodo Tolva) y la operación P (nodo P). El código de la Tolva se ejecuta asignando verdadero a la etiqueta tgLlenar, y termina de ejecutarse cuando el sensor de lleno se activa ocasionando que el código de la Tolva cambie a verdadero la etiqueta tgLleno, como se indica en el modelo. Con la terminación del recurso tiempo RT2, el coordinador identifica que la fase C ha sido realizada (caja llena).

La fase B, utiliza el recurso tiempo RT3, formado por el equipo banda (nodo Banda) y la operación de alimentación/retiro (nodo A/R). Como se indica en el modelo la ejecución del código de la clase (objeto) banda inicia a través de la etiqueta tgRetirar. Mediante la etiqueta tgRetirado el código de la banda indica que la caja ha sido retirada de la banda. De tal manera el sistema genérico (coordinador), interpreta al modelo para la operación de cada una de las funciones en el código de los diferentes objetos (clase Banda y Tolva) para la realización de la operación requerida, en este caso colocar, llenar y retirar.

A manera evaluar las modificaciones requeridas en el programa para un cambio en el sistema a controlar, se trabajó con un cambio en el sistema que consiste en agregar una tolva al sistema original, Fig. 6. El modelo iMRP de tal sistema se muestra en la Fig. 9. Es importante destacar que tal modelo es una abstracción realizada en un editor gráfico iMRP, que genera el modelo en forma de objetos de manera interpretable por el sistema genérico (coordinador).

El modelo es diferente, la lista de fases a realizar son A, B, C y D, requiere los recursos tiempo adicionales RT4 y RT5, así como la representación de la Tolva 2 en el grafo Célula, mediante el nodo Tolva2. El coordinador identifica (interpreta) que la primera fase a realizar es la fase D. Cada fase se encuentra asociada a uno o varios recursos tiempo. Cada uno de los recursos tiempo es iniciado cuando el coordinador establece a verdadero el valor de la etiqueta correspondiente al código del equipo identificada en el atributo del arco entre recurso tiempo y actuador. Así mismo, el fin de la ejecución de un recurso tiempo es indicado en el valor verdadero de la etiqueta correspondiente identificada en el arco entre recurso tiempo y el nodo en el grafo operación. Así, ante el cambio realizado en el sistema a controlar (tolva 2 agregada) el sistema genérico resulta sin modificaciones al interpretar el modelo grafico iMRP del nuevo sistema.

5. CONCLUSIONES.

La técnica iMRP permite abstraer la lógica (o dinámica) de operación del programa así como también la estructura del sistema en cuanto a la ubicación de sensores, actuadores y código de implementación de las clases correspondientes. Las

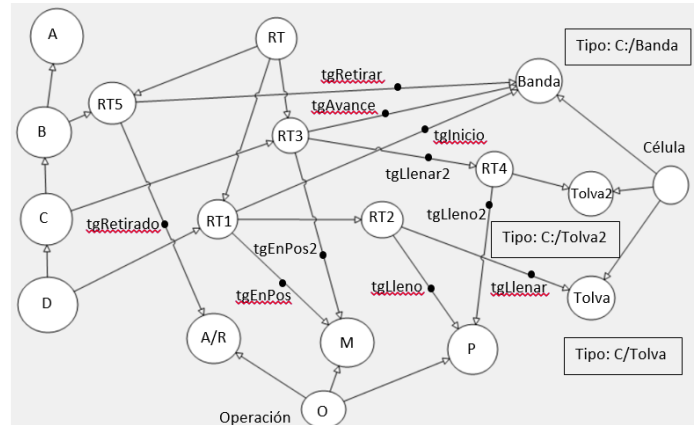


Fig. 9. Modelo iMRP del sistema de envasado modificado (dos tolvas).

técnicas de modelado tradicionales representan en un diagrama únicamente el aspecto lógico o el estructural del sistema. La integración de ambas características del automatismo (estáticas y dinámicas) en un solo modelo de particularidades, facilita la operación dirigida por modelo del sistema genérico. Debido a las características propias de las plataformas de desarrollo actuales para programas de PLC, la solución aquí propuesta requiere la ejecución del sistema genérico en PC. Como trabajo futuro de investigación se plantea la implementación del sistema genérico a operar en el PLC, en donde la maduración de las plataformas de desarrollo respecto al soporte de características de programación orientada a objeto sería un apoyo bastante considerable.

REFERENCIAS.

- [1] L. Racchetti, C. Fantuzzi, L. Tacconi, M. Bonfe, "The PLC UML State-chart Desing Pattern", Emerging Technology and Factory Automation (ETFA), 2014 IEEE , 16-19 Sep. 2014.
- [2] F. Basile, P. Chiacchio, D. Gerbasio, "Progress in PLC Programming for Distributed Automation Systems Control", 2011 9th IEEE International Conference on Industrial Informatics, Caparica, Lisbon, Portugal, 26-29 July 2011.
- [3] J. Desel, W. Reisig, "The Concepts of Petri nets", Software & Systems Modeling, 7 Ago 2014.
- [4] R. L. Marichal y E. J. González "Ullsimgraf: An educational tool whit syntax control for Grafcet notation", Computer Applications in Engineering Education, 16 Feb 2012
- [5] S. Lohmann, S. Engell "Systematic Logic controller Design as Sequential Function Chart Starting form Informal

- Specifications” Chinese Journal of Chemical Engineering Volume 16, Inssue1, Feb 2008, pp 43-47
- [6] J. E. Acosta, F. Sastrón, “Production Shop Floor System Modeling With Reusability Focus”, Cost Effective Automation in Networked Product Development and Manufacturing, 2007.
- [7] Martin Obermeier, Steven Braun, and Birgit Vogel-Heuser, Senior Member, IEEE, ” A Model-Driven Approach on Object-Oriented PLC Programming for Manufacturing Systems with Regard to Usability”, IEEE Transactions on Industrial Informatics, Volume: 11, Issue: 3, June 2015, pp 790 - 800
- [8] Nakagawa, E. Y., Becker, M., Maldonado, J. C. Towards a Process to Design Product Line Architectures Based on Reference Architectures. In *Proceedings of the 17th International Software Product Line Conference*. 2013. p.p. 157–161.
- [9] Acosta, J., F. Sastrón (2006), Schematic Architecture: Reference Architecture / Frameworks / Particular Models for the Shop Floor Environment, 32nd Annual Conf. of the IEEE Industrial Electronics Society, París France, November.
- [10] Acosta, J. F. Sastrón, (2007), Production shop floor system modeling with reusability focus, IFAC Conference on Cost Effective Automation in Networked Product Development and Manufacturing, Monterrey N.L., México, October 2-5.
- [11] Acosta, J. E., Esquema de Referencia para acoplamiento débil entre Sistema informático y equipo de producción, Director: Francisco Baguena, Tesis Doctoral Universidad Politécnica de Madrid, 2016.
- [12] Rodríguez-Rodríguez, D., Acosta-Cano, J. (2017). aplicación del paradigma orientado a objetos en la reusabilidad de código en controladores lógicos programables. Revista Indizada: ISSN: 1405-2172. Congreso de Internacional de Ingeniería Electrónica. Chihuahua, Chih., Octubre 2017, Vol. 39. No. 1. pp. 1-6.