

Propuesta de un sistema de programación y control distribuido para sistemas de fabricación tipo Job Shop

Araúzo Araúzo, José Alberto¹, de Benito Martín Juan José², Sanz Angulo Pedro³, del Olmo Martínez Ricardo⁴

¹ Ingeniero Industrial, ETSII. de la UVA, Paseo del Cauce s/n 47011 Valladolid, arauzo@eis.uva.es

² Dr. Ingeniero Industrial, ETSII. de la UVA, Paseo del Cauce s/n 47011 Valladolid, debenito@eis.uva.es.

³ Dr. Ingeniero Industrial, ETSII. de la UVA, Paseo del Cauce s/n 47011 Valladolid, psangulo@eis.uva.es.

⁴ Dr. Ingeniero Industrial, EPS. de la UBU, Avenida Santander s/n 09006 Burgos, rdeolmo@ubu.es.

RESUMEN

La incorporación de sistemas software como sistemas de información en la empresa es una realidad más que evidente. En la actualidad muchas de las actividades se realizan mediante estos sistemas, de forma que los flujos de información se automatizan e integran. El sistema productivo como subsistema empresarial que es, también dispone de estas tecnologías. Sistemas informáticos como los MRP, CAD, CAM, CAE automatizan e integran ciertas actividades referentes a la fabricación y el diseño. Aun así, existe un aspecto que hasta la actualidad se resiste bastante a este tratamiento: la programación y el control de operaciones en los sistemas de fabricación tipo "Job Shop". Estos sistemas recopilarían información sobre las necesidades de producción, las características del proceso y el estado de los recursos productivos. Con todo ello debería enviar instrucciones a los recursos para satisfacer las necesidades planificadas. La forma tradicional de abordar este problema se ha basado en arquitecturas centralizadas, método que en la práctica ha resultado bastante infructuoso. En la actualidad se está trabajando en propuestas distribuidas. Nosotros proponemos en este artículo una propuesta de este tipo para la programación y control de las plantas productivas tipo "Job Shop".

Palabras claves: Sistema de fabricación, dirección de operaciones, Job Shop, planificación, programación, control centralizado, control distribuido, sistema multiagente.

1. Introducción

Una de las funciones más complicadas de los proceso de fabricación tipo *Job Shop*, es el control de planta o SFC (*Shop Floor Control*). Se corresponde con el conjunto de actividades que se realizan en un sistema de fabricación, orientadas a programar y controlar las tareas que deben ejecutar los diferentes elementos del sistema, para cumplir con la producción planificada. Abarca aspectos como la programación de operaciones a corto plazo (*Scheduling*), supervisión, recogida de datos, control de calidad, control de material en curso, sincronización de máquinas y células, coordinación del transporte, aprovisionamiento de materias primas a las máquinas y células, seguimiento de lotes y de ordenes de trabajo, y mantenimiento de los equipos de la planta.

Varias razones justifican la dificultad del SFC. En primer lugar, muchos problemas de optimización de la fabricación como la programación, son problemas computacionalmente complejos, por lo que en muchas ocasiones el tiempo de cálculo necesario para encontrar la solución óptima va más allá de los límites razonables. En segundo lugar, el Control de Planta es objeto de todo tipo de perturbaciones, que van desde errores del modelo y desviaciones

estocásticas sobre los tiempos esperados de procesamiento, hasta averías de las máquinas. En tercer lugar, los sistemas de fabricación tienden a estar al borde del caos. Sistemas con condiciones iniciales idénticas pueden evolucionar hacia situaciones completamente distintas.

El SFC afecta al funcionamiento del sistema como un todo, de tal forma que mejoras relativamente pequeñas pueden dar lugar a grandes ahorros en el funcionamiento global del sistema. Una de las formas de mejorar este funcionamiento es mediante sistemas informáticos que realicen todas las funciones del SFC y las integren con el resto de actividades del sistema productivo. En la actualidad pocos sistemas de fabricación “*Job Shop*” poseen tales sistemas. La alta variedad de configuraciones “*Job Shop*”, la continua evolución estructural de estos y la propia dificultad del problema, hacen que sea bastante difícil el desarrollo de sistemas universales fácilmente adaptables a las particularidades de cada planta. Ante estas dificultades se han propuesto nuevas arquitecturas de control de planta que mejoran la adaptabilidad y flexibilidad de los sistemas SFC.

2. Intercambio de información en el SFC

Las entradas del SFC son fundamentalmente órdenes de producción, información técnica del proceso y estado actual de la planta (figura 1). Las órdenes de producción representan los pedidos que algún subsistema de planificación táctica emite a planta para que sean elaborados. Los sistemas de planificación táctica emiten estas órdenes con el fin de satisfacer las necesidades de producción y dependiendo de entornos de fabricación pueden ser sistemas informáticos como el MRP o personas experimentadas. Las órdenes de producción pueden contener diferentes datos pero es bastante común que estas se compongan de la siguiente información: ítem (producto o componente) a fabricar, la cantidad que hay que producir y el plazo de entrega.

Para poder procesar las órdenes, el SFC deberá disponer de la suficiente información técnica relativa a los procesos de fabricación. Esta consiste fundamentalmente la descripción de las operaciones de transformación y montaje necesarias para elaborar los ítems, las restricciones en el orden de ejecución de las operaciones (relaciones de precedencia), las materias primas y los componentes necesarios. Además, esta información puede ir acompañada de aspectos adicionales, como pautas de control o normas de almacenamiento, que dependerán en todo caso del sistema de fabricación que se esté tratando.

Otro flujo de información que entra en el sistema SFC es la que se refiere al estado del sistema. Para programar y controlar las actividades de los elementos de la planta, el SFC necesitara tener actualizada la información referente a la situación de los diferentes elementos que va a controlar. Esta información puede incluir datos sobre la disponibilidad y estado de las máquinas, operarios, transportes, ocurrencia de averías, falta de materia prima, etc., así como sobre los eventos que sucedan en lo diferentes elementos (finalización de actividades, roturas, etc.).

El sistema SFC debe procesar toda esta información y generar un conjunto de datos que serán enviados al entorno. Se pueden distinguir dos tipos de salidas: las instrucciones que se envían a los elementos de la planta y la información sobre el cumplimiento o no de objetivos que tiene por destino los subsistemas de planificación táctica. La forma de procesar esta información y generar las salidas está muy ligada con la configuración de la planta, pero en cualquier caso se realizará buscando la consecución de los planes y el máximo aprovechamiento de los recursos.

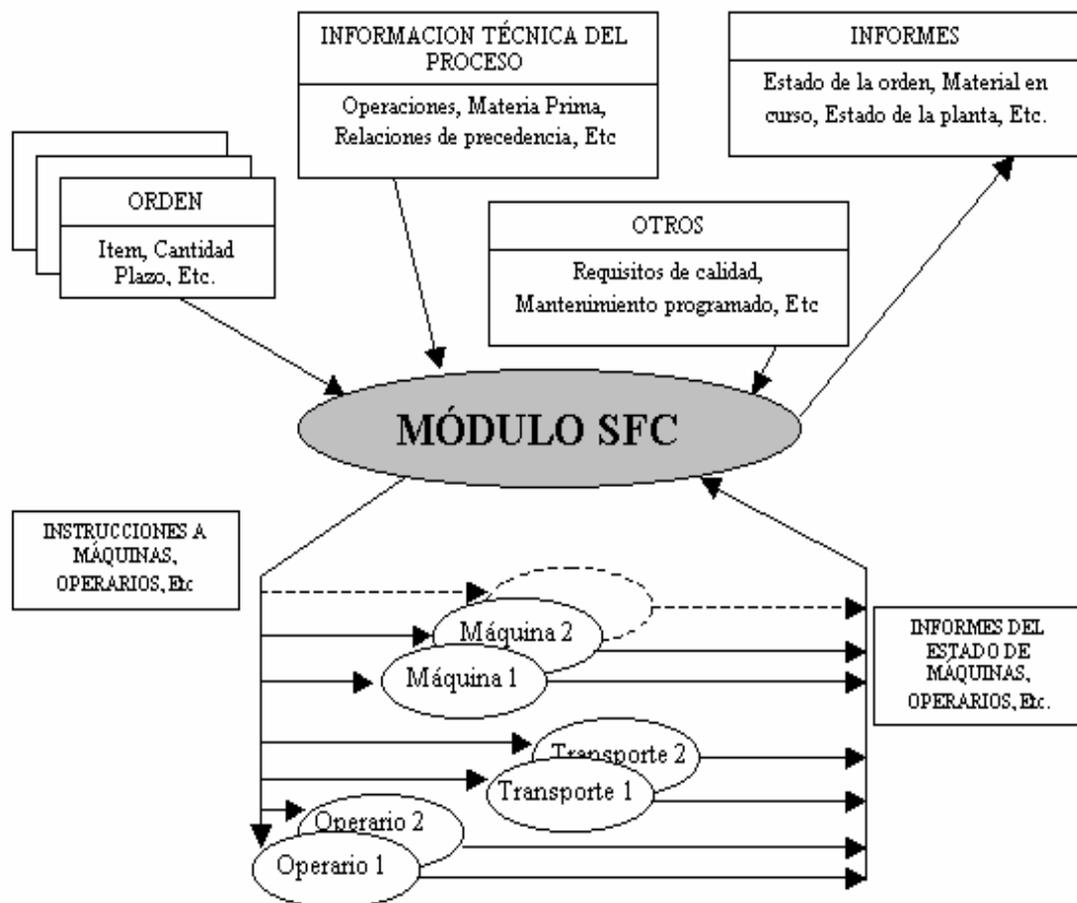


Figura 1. Intercambio de información del sistema de control de planta con el entorno.

3. Funciones del sistema de Control de Planta

Con el objetivo de fabricar las órdenes de producción con costes mínimos, el SFC buscará fabricar los productos dentro de plazo, manteniendo el menor volumen de inventarios posible y utilizando un mínimo de recursos. Aunque como ya se ha dicho, la forma de realizar este cometido depende de la configuración de la planta, en el caso más general debe realizar dos funciones básicas: programación de operaciones y control de la fabricación en tiempo real.

3.1 Programación de operaciones

La programación detallada consisten en la elaboración de un calendario donde se refleje que tareas se van a realizar sobre los distintos pedidos, en que momento y en que centro de trabajo o máquina. Los programas suele tener horizonte temporal de semanas o días y un nivel de detalle que está entre días y minutos.

Generalmente existen muchos programas posibles, pero no todos ellos conducen al cumplimiento de objetivos. Esto hace que el establecimiento de estos calendarios tenga una marcada influencia en la eficiencia del sistema. La dificultad de la programación de

operaciones va a depender de la configuración del sistema de fabricación con el que se esté trabajando, pero el caso más complejo, que es el de fabricación intermitente tipo “Job Shop”, habrá que realizar tres tareas:

- Asignación de operaciones a máquinas o centros de trabajo, indicando que operaciones se realizarán en cada uno de ellos (asignación).
- Establecimiento de la secuencia de tareas que debe realizar las máquinas, para que en la medida de lo posible se cumplan los plazos (secuenciación) y se utilicen los recursos eficientemente.
- Determinación de los momentos de comienzo y finalización de las actividades que debe realizar cada máquina o centro de trabajo, según la secuenciación realizada en el paso anterior (temporización o *scheduling*).

3.2 Control en tiempo real

El Control de la fabricación en tiempo real supone entre otras cosas, la obtención de información sobre el estado de los elementos productivos, para poder lanzar (*dispatching*) o no el programa detallado de producción obtenido anteriormente. El programa se lanzará enviando instrucciones a los elementos productivos. Además se deberá rastrear la evolución de los pedidos y comprobar si se está cumpliendo o no el programa con el fin realimentar el sistema. Dentro de las actividades del control de la fabricación se pueden incluir otras de menor relevancia, como informar de tiempos empleados por los elementos del sistema, monitorizar la planta, realizar el control de calidad, calcular los costes de fabricación, etc.

4. Sistemas de control de planta distribuidos basados en agente

La propuesta que más atención está recibiendo por parte de los investigadores, para el desarrollo de sistemas de control de planta, es el uso de arquitecturas distribuidas basadas en productos y recursos. Según esta idea, el sistema de control debe descomponerse en entidades interrelacionadas que se correspondan con los productos y los recursos que aparecen en el sistema. Esta estructura además de ser sencilla y transparente, es flexible, escalable y fácilmente adaptable, ya que posee la misma estructura física del sistema [Araújo et al (2004)].

La tecnología de agentes software aporta los métodos y herramientas para la materialización de estas ideas. Las órdenes de producción estarán representadas por agentes, que mediante negociación con los agentes que representan a los recursos, establecerán los programas de producción, adaptándolos en cada momento al estado del sistema (figura 2). La cuestión que es motivo de sucesivas investigaciones radica en cómo se debe establecer la relación entre los diferentes agentes para lograr este cometido.

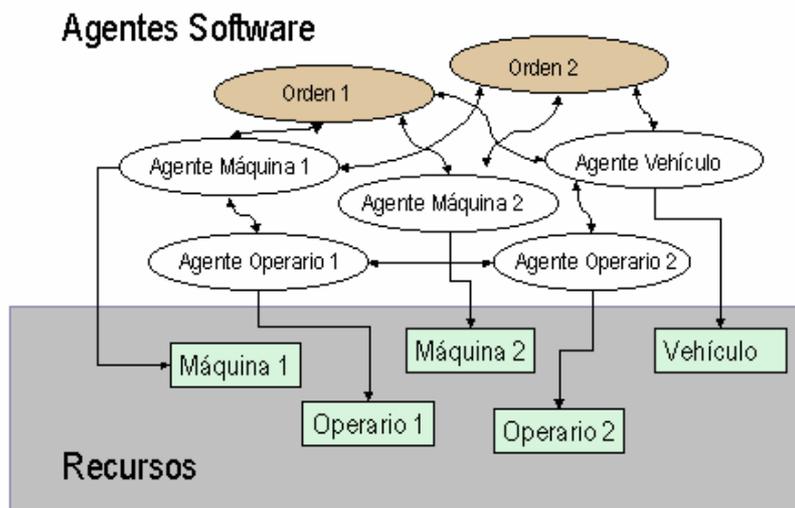


Figura 2. Sistema multiagente para el control de planta.

4.2 Modelos de interacción entre agentes

A la hora de establecer las relaciones entre “agentes orden” y “agentes recurso”, las posibilidades son numerosas. Después de una revisión realizada sobre la literatura del tema nosotros diferenciamos claramente dos tipos de sociedades de agentes: sistemas con agentes programadores y sistemas sin agentes programadores. La existencia o no de este tipo de agentes marca el carácter jerárquico del sistema y por lo tanto las relaciones entre sus componentes [Araújo et al. (2004)].

4.2.1 Sistemas con agentes programadores

Estos modelos se caracterizan por la inclusión de un agente funcional en el sistema, que realiza la programación de operaciones. En estos sistemas los “agentes orden” solicitan al “agente programador” su fabricación. Este último solicita a los “agentes recurso” datos sobre sus capacidades y disponibilidad, y con ellos elabora un programa de fabricación. Según este programa, se envían las diferentes instrucciones a los “agentes recurso” para que operen.

Los sistemas de fabricación están sometidos a continuos imprevistos que invalidan rápidamente los programas realizados. En algunos casos los efectos del imprevisto pueden no justificar una reprogramación, o puede que con pequeños ajustes realizados por los “agentes recurso” del sistema sea suficiente para continuar con las actividades. Para que esto sea posible, estos agentes deben poseer cierta autonomía sobre el agente programador. Las instrucciones de este último serán solo un consejo, que en algunas ocasiones será conveniente seguir y en otras no.

4.2.2 Sistemas sin agentes programadores

En estos, el sistema no se estructuran según una jerarquía y el programa de actividades emerge de la interacción entre los “agentes orden” y los “agentes recurso”. El diseño de estas interacciones entraña una dificultad: hay que conseguir, el cumplimiento de objetivos globales. Además serán preferibles aquellos esquemas que cumplan los siguientes criterios [Baker (1998)]:

- Distribución balanceada: serán preferentes aquellos mecanismos de interacción que distribuyan las necesidades de cálculo de forma equitativa entre diferentes agentes.
- Correspondencia física: los algoritmos que distribuyan el cálculo en entidades que se correspondan con elementos físicos serán más adecuados. Esto se corresponde con, la idea de descomposición del sistema según ordenes y recursos. De esta forma se consigue además, que las necesidades de cálculo crezcan a la vez que el número de agentes del sistema.
- Crecimiento escalable: al crecer el número de elementos del sistema aumenta las posibilidades de secuenciación de las actividades, por lo que también aumenta las necesidades de cálculo. Los algoritmos ideales serán aquellos, en los que el crecimiento de las necesidades de cálculo sea proporcional al crecimiento del sistema, de forma que la cantidad de cálculo por agente es siempre constante. Este será un criterio de difícil cumplimiento.

Para diseñar esquemas de interacción distribuidos se puede partir de los algoritmos de programación existentes y ver como se pueden adaptar estos a una computación distribuida que cumpla con las características anteriores. Baker evalúa los algoritmos respecto a esos criterios clasificándolos en tres categorías: algoritmos de lanzamiento (*dispatching*), algoritmos de secuenciación (*scheduling*) y algoritmos de arrastre (*pull*).

Algoritmos de lanzamiento (*dispatching*). Consisten en un criterio de decisión mediante el cual se selecciona una operación, entre las que quedan por ejecutar, para ser realizada mediante un recurso disponible. Una vez tomada la decisión, no se vuelve a reconsiderar, si no que se ejecuta la operación, modificándose de esta forma la disponibilidad de recursos y el conjunto de operaciones que queda por fabricar. Los diferentes algoritmos de lanzamiento se caracterizan por el criterio de decisión utilizado para seleccionar operación y recurso: se puede seleccionar la operación más urgente, la de mayor duración, la máquina menos utilizada, etc. Estos algoritmos cumplen bastante bien los tres criterios de evaluación anteriores y además se pueden implementar fácilmente en sistemas multi-agentes no jerárquicos. Dependiendo del algoritmo usado se pueden seleccionar diferentes formas de implementación, pero todas ellas se basan en un esquema de dialogo entre agentes software conocido como “*contract-net protocol*”. O bien los agentes producto piden ofertas a los agentes máquina, para luego seleccionar la más adecuada, o bien los agentes máquina solicitan trabajo a los agentes orden y así seleccionar la operación que prefieren realizar [Saad et al. (1996)]. Son algoritmos simples y rápidos pero tienen la desventaja de no asegurar secuencias de fabricación muy eficientes. Su uso esta recomendado en sistemas muy dinámicos, donde los pedidos llegan de forma aleatoria y la disponibilidad de las máquinas es bastante impredecible.

Algoritmos de secuenciación (*scheduling*). Aunque en algunas ocasiones se usa este término para designar a cualquiera de los tres tipos de algoritmos, Baker lo usa sólo para referirse a aquellos métodos de programación en los cuales se realiza un programa completo antes de comenzar a ejecutar ninguna operación. Esta forma de proceder se justifica por la conveniencia de buscar un programa de suficiente calidad antes de comenzar a realizar operaciones. Como es bien conocido, realizar programas óptimos es un problema en NP-Completo, por lo que en muchas ocasiones es inviable y se recurre a métodos que obtienen programas de razonable calidad, pero sin asegurar el óptimo. Existen tantos algoritmos de secuenciación como métodos de optimización combinatoria. Dentro de estos algoritmos se

tiene algunos que buscan el óptimo (programación matemática, búsqueda combinatoria, etc.) y otros que encuentran soluciones aceptables aunque no tiene por que ser óptimas (métodos heurísticos, meta-heurísticos, relajación lagrangiana, etc.). La mayoría de estos métodos son de difícil implementación en sistemas multi-agentes no jerárquicos, ya que por lo general es necesario disponer de un conocimiento completo del problema para poder buscar la solución e ir mejorándola. Sin embargo algunos métodos muy prometedores han sido propuestos en ciertos trabajos, para implementar mecanismos de negociación entre productos y recursos en sistemas multi-agentes. Entre estos destacan, los métodos de relajación lagrangiana en los que la secuencia de fabricación emerge como resultado de un proceso de negociación entre productos y recursos [Kaskavelis et al. (1998)].

Algoritmos de arrastre (*pull*). Estos métodos se basan en un sistema de arrastre de inventarios. Los pedidos solo son comunicados al centro de trabajos que realiza la última operación del producto y este comunica sus necesidades de materiales a los centros de trabajo que le suministran piezas. Estos centros de trabajo hacen lo mismo con los centros de trabajo que le preceden. Continuando con este proceso, las necesidades de producción se trasladan desde el último centro de trabajo hasta el almacén de materias primas. Este método, implementado mediante un sistema de información basado en tarjetas llamado “*Kanban*”, es la base de la programación de tareas en los sistemas JIT (*Just In Time*). Su inconveniente está en que fue ideado para funcionar en un sistema productivo de características determinada, por lo que no se adapta muy bien a todo tipo de configuraciones “*Job-Shop*” ni a sistemas fabricación flexibles o ágiles.

5 Esquema de interacción propuesto

Nuestra propuesta de interacción se basa en la superposición de dos mecanismos. Uno de ellos se basa en el “*Contract-Net Protocol*” mediante el cual las ordenes acuerdan con las máquinas la realización de trabajos. El otro es un mecanismo de negociación basado en precios, por el cual se establecen unos precios de mercado, para el uso de los recursos en diferentes momentos. Estos dos mecanismos se influirán mutuamente. Por un lado los acuerdos logrados mediante el “*Contract-Net Protocol*” modifican las necesidades de trabajos y la disponibilidad de recursos con los que negocian las órdenes y máquinas en el mecanismo de mercado. Por otro lado los precios logrados en el mecanismo de mercado influyen en los acuerdos que se realizan mediante el “*Contract-Net Protocol*”.

En la figura 3 se puede observar como se realiza el “*Contract-Net Protocol*” entre órdenes y máquinas (suponiendo que las máquinas son los únicos recursos). Las ordenes (orden 1 y orden 2) solicitan a las máquinas ofertas para la realización de una operación (cfp). La operación se selecciona de entre todas aquellas que no están programadas y que según las relaciones de precedencia se pueden programar. Las máquinas en función de su disponibilidad que es consecuencia de los acuerdos ya realizados (Plan de máquina) envían una propuesta a las ordenes (proposal) o se rechaza la petición (refuse). Si la propuesta de la máquina es aceptable para la orden esta la aceptará (accept-proposal) y modificará su estado añadiendo la operación acordada al conjunto de operaciones programadas. Cuando la máquina recibe la notificación de aceptación añadirá este nuevo acuerdo a su programa de operaciones.

La cuestión que surge al implementar este mecanismo es como toman los agentes las diferentes decisiones implicadas en el proceso (¿Qué petición de oferta debe atender la máquina?, ¿Cómo acepta o rechaza cada orden la propuestas de las máquinas?, Etc.). Lo más común es tomar estas decisiones en base a criterios basados en reglas de lanzamiento

Nosotros proponemos dos mecanismos de negociación que se superponen, influyendo el resultado del uno en el comportamiento del otro. Un mecanismo basado en el “*Contract-net Protocol*” asegura la correcta coordinación entre elementos, así como una respuesta inmediata a cualquier evento inesperado. Simultáneamente un mecanismo paralelo de negociación basada en precios aporta información al agente, sobre necesidad de los recursos limitados. Esta información es utilizada para tomar las decisiones en las diferentes etapas del “*Contract-net Protocol*” de forma que sea más eficiente el uso de los recursos.

Referencias

Araújo J.A, De Benito J.J, Sanz. P., Del Olmo R, (2004), “Situación Actual y Expectativas de los Sistemas de Fabricación Basados en Agentes”, *VIII Congreso de Ingeniería de Organización Leganes, 9-10, Septiembre 200*.

Baker A.D. (1998),. “A Survey of Factory Control Algorithms which Can be Implemented in a Multi-Agent Heterarchy”, *Journal of Manufacturing Systems*, Vol. 17, No. 4, 1998, pp. 297-320.

Saad A., Kawamura K., Biswas G. (1996) “Performance Evaluati6n of Contract Net-Based Heterarchical Scheduling for Flaxible Manufacturing Systems”, *Intelligent Autonomous and Soft Computing*, 3(3), pp 229-248.

Kaskavelis C.A., Caramanis M.C., (1998) “Eficient Lagrangian relaxation algorithms for industry size job-shop scheduling problems”. *IIE Transactions*, Vol 30, pp 1085-1097.