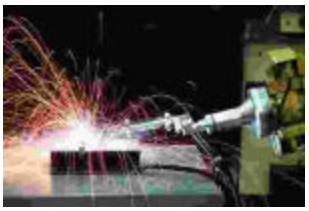
Ingeniería concurrente y tecnologías de la información

Rodolfo García Flores
Profesor del Doctorado en Ingeniería de Sistemas de la FIME-UANL.
rodolfo@yalma.fime.uanl.mx



RESUMEN

La ingeniería concurrente (CE por sus siglas en inglés) es un enfoque para la manufactura que permite el diseño y desarrollo simultáneo de productos, procesos y actividades de apoyo. Aunque éste no es un concepto nuevo, ha recibido recientemente cierto empuje de tecnologías de la información como el Internet o algunas técnicas de Inteligencia Artificial. Específicamente, el uso de agentes de software y lenguajes para el manejo de conocimiento pueden aportar una base confiable y flexible para el desarrollo de plataformas de ingeniería concurrente. Este artículo presenta una introducción a los conceptos relacionados con CE, el papel que los agentes de software y el modelado de datos juegan en ella, y describe brevemente un caso de estudio.

PALABRAS CLAVE

Ingeniería concurrente, agentes computacionales, inteligencia artificial, tecnología de la información.

ABSTRACT

Concurrent Engineering (CE) is a systematic manufacturing approach that allows parallel design and development of products, related processes and support activities. Although this is not a new concept, it has received a boost from newly developed information technologies like the Internet and tools derived from Artificial Intelligence. Specifically, the use of software agents and knowledge manipulation languages can provide a reliable and flexible basis for CE platform development. This paper presents an introduction to concepts involved in CE, the role that software agents and data modelling are playing on it, and describes a CE project built upon software agents.

KEYWORDS

Concurrent engineering, software agents, Artificial Intelligence, information technology.

INTRODUCCIÓN

La ingeniería concurrente (CE por sus siglas en inglés) se define como "un enfoque sistemático para el diseño paralelo e integrado de productos y los procesos relacionados, incluyendo manufactura y servicios de apoyo, con la intención de que los desarrolladores consideren, desde el inicio del proyecto, todos los elementos del ciclo de vida del producto, desde su concepción hasta su eliminación y reciclaje, incluyendo calidad, costo, planeación y requerimientos del usuario".[1] Cuando se implementa exitosamente, los productos que se desarrollan con esta filosofía se fabrican de forma eficiente, entran al mercado rápidamente y son de calidad satisfactoria para los clientes.

El término CE se ha venido usando desde 1986, cuando el Instituto para el Análisis de la Defensa de Estados Unidos lo describió en su reporte R-388.[2] Hoy ésta es un área de investigación muy lucrativa. CE mejora el enfoque secuencial de la producción tradicional mediante tres elementos principales:

- Una arquitectura computacional distribuida que permite la sincronización, la programación óptima de tareas y el manejo adecuado de flujos de información.
- Una representación unificada de toda la información de diseño y manufactura, de forma que pueda visualizarse e interpretarse desde diversas perspectivas.
- Un conjunto de herramientas computacionales que permiten desarrollar prototipos a bajo costo, de forma óptima e inteligente.

La diferencia entre ambos enfoques puede apreciarse en las figuras 1 y 2. Aun cuando en el enfoque secuencial es posible volver a las fases anteriores de desarrollo del producto, las tareas deben realizarse una a la vez. En cambio, el enfoque concurrente permite la realización simultánea de todas las tareas de desarrollo hasta la fabricación del prototipo. Otros conceptos que distinguen a CE del enfoque tradicional son el cambio en cultura organizacional, los equipos de trabajo multidisciplinarios y el énfasis en el manejo de rutas de información más que de jerarquías organizacionales.

Aunque el concepto no es nuevo, el desarrollo reciente de tecnologías de la información como Internet y ciertas técnicas de Inteligencia Artificial permite crear nuevas aplicaciones para explotar mejor la filosofía de la ingeniería concurrente. En el presente artículo se explica en particular el papel que están jugando dos de estas herramientas para el avance de CE: los agentes computacionales y la modelación de datos. La descripción breve de un proyecto realizado para este fin complementa la exposición.

AGENTES COMPUTACIONALES

Los grupos de trabajo multidisciplinarios –deseables para desarrollar productos en paralelo según el enfoque de la ingeniería concurrente– poseen capacidad de decisión, responsabilidades y cierta libertad para manejar sus propios recursos. Además puede suceder que físicamente el personal se encuentre localizado en diferentes ciudades o países. Con equipos de personas de estas características es natural que el trabajo se realice a través de redes de cómputo utilizando entidades que posean cierta autonomía para representar a los distintos grupos y que sean capaces de comunicarse entre sí.

Los agentes computacionales (o de software) pertenecen a una rama de la Inteligencia Artificial conocida como Inteligencia Artificial Distribuida, y aunque no existe una definición unánimemente aceptada, se reconoce que éstos son programas que funcionan de forma autónoma o semiautónoma y que están en comunicación con otros agentes, humanos o computacionales.[3] El concepto de autonomía expresado en esta definición implica que, a diferencia

Fig. 1. El enfoque secuencial de la manufactura.[2]

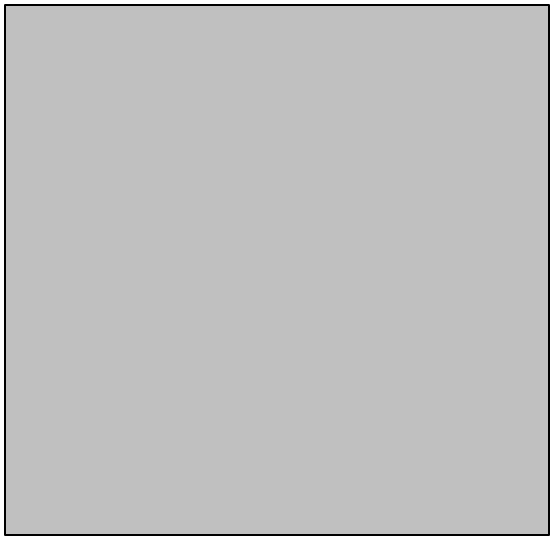


Fig. 2. El enfoque de la ingeniería concurrente para la manufactura.[2]

de los programas convencionales, los agentes poseen intereses e iniciativa propios para acometer acciones sobre su ambiente. Otras facultades que se les puede conceder, dependiendo de su conveniencia para proyectos específicos, son la comunicación y el aprendizaje. Los agentes han demostrado ser una técnica útil para diseñar sistemas distribuidos y cooperativos en muchas actividades industriales y de servicios, incluyendo las telecomunicaciones, el control de tráfico aéreo, la administración del transporte, el cuidado médico y el entretenimiento.[4] Todas estas propiedades (autonomía, distribución geográfica, cooperación, aprendizaje y comunicación) hacen que los agentes de software sean ideales para el desarrollo de aplicaciones en CE. No obstante, para lograr estas propiedades es necesario alcanzar primero un flujo efectivo de información entre los agentes participantes. Existen tres condiciones para ello:

1. Una ontología compartida. – Los agentes deben tener una misma visión del mundo, o en términos coloquiales, un vocabulario compartido.
2. Un protocolo o lenguaje común. – Todos los agentes deben ser capaces de comprender el lenguaje utilizado por los demás para intercambiar mensajes.
3. Un formato común para el contenido de la información. – El contenido de los mensajes mismos debe ser interpretable por todos los participantes.

A manera de ilustración, imagínese que un ambiente de CE un agente A se utiliza para manipular aplicaciones ya desarrolladas (por ejemplo, un programa para CAD/CAM) y comunicar el contenido de archivos de estas aplicaciones a otros agentes en términos que éstos últimos puedan entender (requerimiento 1). Supongamos que el agente A solicita al agente B la corrección del archivo CADX. La solicitud de revisión se codifica en un lenguaje común (requerimiento 2), mientras que el contenido del mensaje se codifica en el formato común (requerimiento 3). El mensaje original posiblemente tiene un formato específico al programa de diseño asistido por computadora que el agente "A" manipula, pero si los agentes en el sistema cumplen los requerimientos arriba mencionados, el contenido puede ser compartido y transformado de forma inteligente por todos los participantes de la plataforma de ingeniería concurrente. La siguiente sección ahonda en los requerimientos de modelado de datos mediante el formato común.

MODELACIÓN DE DATOS

Los grupos de desarrollo de productos frecuentemente están integrados por personas que provienen de distintos contextos profesionales y no comparten el mismo vocabulario técnico. Además requieren trabajar con paquetes de software que muchas veces no son compatibles entre sí, o pueden existir fases del proceso de desarrollo que no están completamente automatizadas y requieren captura manual de datos. Esta falta de consistencia en los sistemas de información produce retrasos y gasto innecesario de recursos en traducciones, y disminuye el valor de los datos para la empresa.

El objetivo último del manejo de datos es hacer que la información adecuada esté a disposición del personal adecuado en el momento adecuado. Para lograrlo, las estructuras de datos deben desarrollarse de forma que sean claras, accesibles, consistentes, completas, relevantes y precisas.

Esta no es una dificultad que haya surgido junto con CE. El intercambio de datos siempre ha sido un problema para la industria y ha producido diversas iniciativas,[5] como por ejemplo EDIF (2000), IGES (1991), y GKS (1985). Sin embargo, existen pocas iniciativas pensadas 1) para el intercambio de información en redes, 2) que sean consistentes con todas las actividades relacionadas con el desarrollo de proyectos además de CAD/CAM, y 3) que tengan formato neutral, esto es, que evite el sesgo hacia una aplicación en particular. Aquí mencionaremos dos lenguajes de modelación de datos que cumplen estos requisitos: EXPRESS y XML.

- EXPRESS es el lenguaje de modelación de datos de STEP (Standard Exchange of Product model data). STEP se creó para ser el estándar internacional para el intercambio de información en manufactura y se aprobó como ISO 10303 en 1987. A pesar del importante esfuerzo dedicado a crear sus protocolos de aplicación (ontologías) y actualizarlo, el lenguaje no ha sido tan aceptado como se esperaba.
- XML ("eXtensible Mark-up Language") es un lenguaje que, aunque no es en sí mismo un estándar de manufactura, sí es un lenguaje de modelación de datos con el que se están desarrollando actualmente diversos estándares para varias aplicaciones (véase por ejemplo, http://www.service-architecture.com/xml/articles/xml_vocabularies.html). XML fue creado en 1996 por el Consorcio de la Red Mundial (W3C). Éste es un lenguaje de "margen" de la misma familia que HTML, pero con importantes mejoras: permite al usuario especificar atributos nuevos, admite revisar la validez de los datos modelados y da la posibilidad de crear estructuras de datos. Se espera que en el futuro XML sustituya a HTML como el lenguaje de creación de páginas en Internet.

La tendencia hoy es a aprovechar las ontologías desarrolladas para STEP codificándolas con XML. Para ello existen varios proyectos, como por ejemplo PDML (1999), que es un vocabulario diseñado para el intercambio de información entre los sistemas del Departamento de Defensa de Estados Unidos y sus proveedores. La principal ventaja de utilizar esta combinación es el aprovechamiento de la infraestructura ya existente para Internet para intercambiar datos de manufactura reutilizando el trabajo en estándares de STEP. El proyecto que se describe en la siguiente sección utiliza este enfoque e integra los conceptos expuestos hasta este momento.

UN SISTEMA MULTI-AGENTE

Un sistema de empresas participantes en una cadena productiva de la industria química fue modelado mediante agentes computacionales[5] como prototipo de un ambiente de CE. Cada entidad fue emulada por agentes que tienen la estructura que se muestra en la figura 3. Los módulos que componen los agentes individuales se identifican por sus siglas en inglés en el recuadro gris. Por ejemplo, el modelo que el agente tiene de sí mismo (self model) se identifica como SM, y así sucesivamente. El agente cuenta con una representación de sí mismo (SM) así como de los demás agentes (AM), módulos para administrar los servicios que presta (SEM), para evaluar la situación del ambiente (SAM) y para manejar sus interacciones con otros agentes (IMM). La estructura también contempla un módulo de comunicaciones.

Fig. 3. Estructura de agentes individuales. Cada módulo se identifica por sus iniciales en inglés en el recuadro gris.

La estructura de los agentes individuales se implementó usando el lenguaje Java. Las razones son las siguientes:

- Independencia de plataforma. – Es importante que en un ambiente de trabajo colaborativo las aplicaciones administrativas y de ingeniería puedan comunicarse con mínimo esfuerzo. Los programas escritos en Java no necesitan recopilarse para correr en diferentes sistemas.
- Seguridad. – Es uno de los primeros lenguajes en considerar la seguridad en su diseño.
- Concurrencia. – Permite de forma explícita la programación de tareas paralelas, por lo que es posible trabajar conceptualmente con tareas simultáneas.

Los mensajes entre agentes en el prototipo para CE tienen una estructura estratificada. La figura 4 representa los diferentes niveles de implementación de los lenguajes empleados. Java se encuentra en el nivel más externo, pues es el lenguaje de construcción de los agentes. Los globos representan el nivel del protocolo, esto es, el lenguaje en el que los agentes intercambian mensajes, instrucciones o recomendaciones sobre qué hacer con la información. En la figura se representa con el lenguaje KQML (Knowledge Query and Manipulation Language), que está siendo desarrollado especialmente para los agentes de software.[6] El nivel de contenido representa la información en sí, y se codifica en XML o EXPRESS para aprovechar la infraestructura existente utilizando las ontologías desarrolladas para STEP.

Fig. 4. Niveles de implementación de los lenguajes.

En el prototipo, las aplicaciones manipuladas por distintos agentes tienen como objetivo determinar la secuencia óptima de tareas de manufactura, las rutas óptimas de distribución de materia prima y el manejo de inventarios con políticas adecuadas.[7, 8] El sistema también cuenta con un mecanismo para la resolución de conflictos, cuya descripción detallada puede encontrarse en las referencias citadas.

COMENTARIOS FINALES

La ingeniería concurrente es un enfoque de manufactura que permite el diseño y desarrollo integrado de productos y sus procesos relacionados. CE fomenta el desarrollo de tareas en paralelo, los equipos de trabajo multidisciplinarios y el intercambio eficiente de información. Aunque para la industria estas características han sido deseables y necesarias desde hace tiempo, los avances recientes en tecnología de la información como Internet o ciertas técnicas de Inteligencia Artificial les han dado una solución práctica en años recientes, como atestigua PDML, un vocabulario utilizado para el intercambio de información entre los sistemas del Departamento de Defensa de Estados Unidos y sus proveedores. En particular, los agentes de software y algunas iniciativas para estandarizar la descripción de productos de manufactura como STEP y algunos vocabularios de XML han dado lugar a importantes avances en la aplicación de la ingeniería concurrente. En este artículo se ha ilustrado su potencial con la descripción de un proyecto realizado con agentes de software.

El uso de agentes computacionales y lenguajes para el manejo de conocimiento, junto con nuevas ontologías, puede proveer de una base confiable y flexible para crear plataformas de desarrollo de bajo costo para CE aprovechando la infraestructura desarrollada en los últimos años para Internet.

GLOSARIO

AM – Modelo de los otros (Acquaintance Model).

CE – Ingeniería concurrente (concurrent engineering).

CAD/CAM – Diseño asistido por computadora / manufactura asistida por computadora.

CM – Manejador de comunicaciones (Communication Manager).

HTML - Lenguaje de margen para hipertextos (HyperText Mark-up Language).

IMM – Módulo de manejo de interacciones (Interaction Management Module).

IOM – Manejador de archivos (Input/Output Manager).

KQML – Lenguaje de manipulación de conocimiento (Knowledge Query and Manipulation Language).

PDML – Lenguaje de margen para datos de producto (Product Data Markup Language).

SAM – Módulo de evaluación de situaciones (Situation Assessment Module).

SEM – Módulo de ejecución de servicios (Service Execution Module).)

SM – Modelo de sí mismo (Self Model).

STEP – Estándar ISO-10303 (STandard Exchange of Product model data).

XML – Lenguaje de margen extensible ((eXtensible Mark-up Language)

REFERENCIAS

1. S.L. Albin y P.J. Crefeld. Getting started: Concurrent engineering for a medium-sized manufacturer. Journal of Manufacturing Systems, 13: 48-58, 1994.
2. R. Mills, B. Beckert y L. Carrabine. The future of product development, Computer-Aided Engineering, 10: 38-46, 1991.
3. D. O´Leary, D. Kuokka y R. Plant. Artificial Intelligence and virtual organizations, Communications of the ACM, 40: 52-59, 1997.
4. N.R. Jennings, T.J. Norman, P. Faratin, P. O´Brien y B. Odgers. ADEPT: an agent-based approach for to business process management, ACM Sigmod Record, 27: 32-39, 1998.
5. R. García-Flores. A multi-agent system for chemical supply chain simulation, management and support. Tesis doctoral. Universidad de Leeds, Reino Unido, 2002.
6. Y. Labrou y T. Finin. A proposal for a new KQML specification, documento TR CS-97-03. http://www.csee.umbc.edu/~jklabrou/publications/tr9703.ps, 1997.
7. R. García-Flores, X.Z. Wang y G. Goltz. Agent-based information flow for process industries´ supply chain modelling, Computers and Chemical Engineering 24: 1135-1141, 2000.
8. R. García-Flores y X.Z. Wang. A multi-agent system for chemical supply chain simulation and management support, OR Spectrum 24: 343-370, 2002.