

## **Programación de Actividades de Procesos de Negocio Extendido: PPNE\_AGENT**

**Pedro Gómez<sup>1</sup>, Ángel Ortiz<sup>2</sup>**

Centro de Investigación de Gestión e Ingeniería de la Producción (CIGIP). Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Valencia, Campus de Vera, 46022 Valencia. <sup>1</sup> [pgomez@cigip.upv.es](mailto:pgomez@cigip.upv.es), <sup>2</sup> [aortiz@cigip.upv.es](mailto:aortiz@cigip.upv.es)

### **Resumen**

*En el marco de la ejecución de Procesos de Negocio Extendidos (PNE), es fundamental contar con un sistema capaz de adaptarse a la evolución de los acontecimientos de forma ágil y dinámica. Este trabajo establece las bases para la creación de una herramienta que actúe como programador de actividades dentro de un PNE. El programador tendrá en cuenta los recursos y los eventos disponibles en cada instante, e intentará maximizar la satisfacción del cliente. El trabajo define el modelo matemático estático, así como la arquitectura de un Sistema Multiagente (SMA) y su metodología de diseño para el sistema dinámico.*

**Palabras clave:** Programación, Procesos Negocio, BPMS.

### **1. Introducción**

En los últimos años se están analizando los procesos de negocio de las empresas desde la perspectiva de la Cadena de Suministro o desde la de Red de Empresas. Cuando los procesos de una empresa deben interactuar con los de otras empresas con las que se trabaja en red nos referimos a ellos como Procesos de Negocio Extendido (PNE).

Un aspecto tradicionalmente abordado en este ámbito es el de la Interoperabilidad, es decir, permitir que los procesos de distintas empresas sean capaces de trabajar de forma conjunta aunque se hayan desarrollado y su ejecución esté soportada por herramientas distintas. No obstante, una vez superada esta dificultad, relacionada con la factibilidad de las soluciones propuestas en el marco del diseño de los procesos, y ya en el campo de las operaciones de procesos, no se debe perder de vista la mejora en la eficacia. Este trabajo se centra en el procedimiento de programación de las actividades que componen un proceso extendido.

### **2. Descripción del Problema**

Los procesos de negocio están compuestos por actividades que deben realizarse siguiendo un orden determinado según se establece en las reglas de cada proceso de negocio. Las actividades pueden ser ejecutadas por uno o varios recursos, que deberán estar disponibles de lo contrario las actividades se podrán en cola hasta poder ser atendidas. El número de “ejecutores” puede ser múltiple para cada actividad. Así mismo, existen caminos alternativos en la ejecución del un proceso, pudiendo variar la ejecución de forma dinámica (Vieira, Herrmann, and Lin, 03) en función de un evento

contemplado aunque desconocido hasta un momento dado, por ejemplo no es lo mismo recibir el pago de una factura o una devolución de ésta.

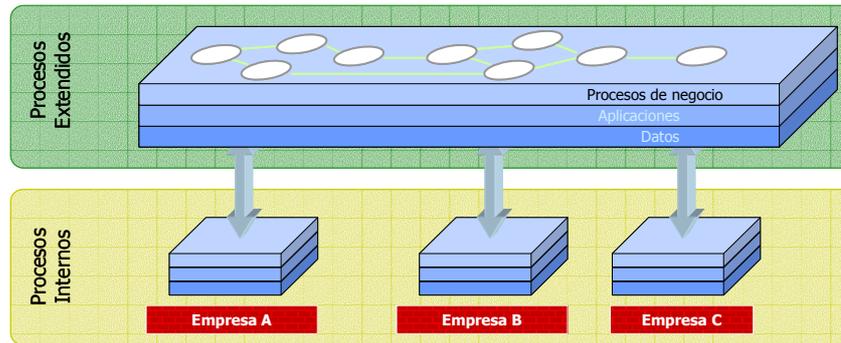


Ilustración 1. Proceso de Negocio Extendido Franco et al. (2001).

Un factor importante en la ejecución de los procesos es la calidad de los recursos con los que se cuenta para su ejecución. Por ejemplo, una actividad que puede ser realizada por varias personas puede obtener diferentes calidades en función de quien la ejecute (experiencia, cansancio, habilidad, etc.).

A demás de las características anteriores en un PNE entran en juego otras que deben ser tenidas en cuenta. Los procesos de Negocio Extendidos son por su propia naturaleza distribuidos y el grado de incertidumbre (Galbraith, 73) de los tiempos de llegada de nuevas tareas y del volumen demandado es alto. Por lo que la programación de actividades de un PNE entra dentro del ámbito la toma de decisiones distribuidas, analizadas en los sistemas complejos (Schneeweiss, 03).

El problema que se plantea es como realizar las diferentes actividades que componen el un proceso de forma exista un balance adecuado entre la calidad en la ejecución de las actividades y el respeto por la fecha de entrega comprometida con el cliente. La solución que se propone se basa en el diseño de una plataforma para la generación de programas factibles con el objetivo de intentar minimizar un parámetro. Esta plataforma, como se explicará más adelante, se integrará en un BPMS (Business Process Management System).

### 3. El Modelo Matemático Estático y Determinista

Como paso previo al planteamiento real, es decir en un entorno dinámico y estocástico, se plantea en este apartado el modelo matemático que facilitará el entendimiento del problema en condiciones menos complejas. El modelo propuesto define las características, requerimientos y objetivos descritos en lenguaje natural en el apartado anterior y se centra únicamente el caso estático y determinista.

Los datos básicos de entrada son:

- ⇒ N: Número de trabajos a realizar durante el periodo objeto de la programación.
- ⇒ R(i): Número de actividades de el proceso i.
- ⇒ M(r): Número de ejecutores en la actividad r
- ⇒ p(i,j,r): Tiempo de proceso del trabajo i en el ejecutor j de la actividad r.

- ⇒  $q(i,j,r)$ : Calidad de servicio del ejecutor  $i$  para la el trabajo  $i$  de la actividad  $r$ .  $q$  estará normalizado entre 1 y 2 de tal forma que 1 será la mejor calidad y 2 la peor calidad.
- ⇒  $Q(i)$ : Calidad media aceptable en el proceso  $i$ .
- ⇒  $d(i)$ : Fecha de finalización del trabajo  $i$ .
- ⇒  $\Omega$ : Un número arbitrariamente grande.

Las variables son:

- Variables Binarias

$$y(i, j, r) \begin{cases} =1 & \text{Si el trabajo } i \text{ es procesado por el ejecutor } j \text{ en la} \\ & \text{actividad } r. \\ =0 & \text{En otro caso} \end{cases}$$

- Variables Continuas

- ⇒  $c(i,r)$  fecha máxima de inicio de las operaciones sobre el trabajo  $i$  en la actividad  $r$ .
- ⇒  $CM(i,r)$  fecha de finalización del trabajo  $i$  en la actividad  $r$ .

Función objetivo:

$$\min \sum_{i=0}^N \left[ \sum_{r=1}^{R(i)} \sum_{j=1}^{M(r)} (q(i, j, r) * y(i, j, r)) \right] * T_i \quad \text{donde } T_i = \max(0, C_i - d_i)$$

Como se ve la función objetivo pretende mejorar la atención del cliente posibilitando la realización de las actividades balanceando los conceptos de calidad en la ejecución la misma y fecha de entrega.

Las restricciones son:

$$\text{Ecuación 1 } \sum_{j=1}^{M(r)} y(i, j, r) = 1 \quad i=1 \dots N, \quad \forall r=1 \dots R$$

$$\text{Ecuación 2 } \sum_{i=1}^{N+1} x(0, i, j, r) = 1 \quad \forall j=1 \dots M(r), \quad \forall r=1 \dots R$$

$$\text{Ecuación 3 } \sum_{i=0}^N x(i, n+1, j, r) = 1 \quad \forall j=1 \dots M(r), \quad \forall r=1 \dots R$$

$$\text{Ecuación 4 } c(i_2, r) \geq c(i_1, r) + p(i_1, j, r) + \Omega * \left( \sum_{j=1}^{M(r)} x(i_1, i_2, j, r) - 1 \right) \\ \forall i_1=1 \dots N, \quad \forall i_2=1 \dots N, \quad \forall r=1 \dots R$$

$$\text{Ecuación 5 } c(i, r) \geq c(i, r-1) + p(i, j, r-1) \\ \forall i=1 \dots N, \quad \forall r=2 \dots R$$

$$\text{Ecuación 6 } CM(i) \geq c(i, R(i)) + p(i, j, R(i)) \\ \forall i=1 \dots N, \quad \forall r=2 \dots R$$

$$\text{Ecuación 7 } \left( \sum_{r=1}^{R(i)} \sum_{j=1}^{M(r)} (q(i, j, r) * y(i, j, r)) \right) / R(i) < Q(i) \\ \forall i=1 \dots N, \quad \forall r=1 \dots R$$

Comentarios a las restricciones:

En general se requiere que todos los procesos sean programados en un ejecutor y que cada actividad asociada a un proceso siga un orden de ejecución de forma que la actividad 1 siempre se ejecute antes que la 2. Así mismo se requiere que la fecha de inicio de cada actividad siempre sea posterior a la de fin de la actividad inmediatamente anterior. Siendo la calidad global de un proceso superior a una cota.

#### 4. Propuesta de Plataforma de Programación de Actividades de Procesos de Negocio Extendido

Una vez establecido el modelo estático y entendido el problema es necesario trasladar éste a su entorno natural, es decir a un entorno dinámico sujeto a eventos. Para hacer frente al planteamiento dinámico se ha seguido una metodología de diseño que permite incorporar la visión dinámica desde el inicio en el proceso de diseño, pero también permite reutilizar los conceptos básicos del modelo estático.

Se propone el modelado Multiagente del Sistema para programación de las actividades de los PNE. En concreto se utiliza la metodología INGENIAS (Pavón y Gómez (2003)), la herramienta de modelado INGENIAS IDK 2.3.

Se define una herramienta SMA llamada PNE\_AGENT, que a partir del diseño del PNE y de las reglas de negocio con sus restricciones, obtenga una programación balanceada entre los objetivos de calidad y fecha de entrega, tal y como indica el modelo matemático estático. La programación de cada una de las actividades de un proceso es realizada por un agente que forma parte del SMA. La cooperación entre los diferentes agentes se realiza mediante un protocolo de negociación diseñado al efecto, que está

embebido en el algoritmo que resuelve el modelo planteado. Las ventajas de la solución propuesta son reactividad, distribución y adaptación a las singularidades.

A partir de estas líneas la comunicación se va a centrar en explicar cuales han sido las bases para el desarrollo de la plataforma de programación de actividades de procesos de negocio extendido, como se ha realizado el diseño y cual es la propuesta arquitectónica de la plataforma.

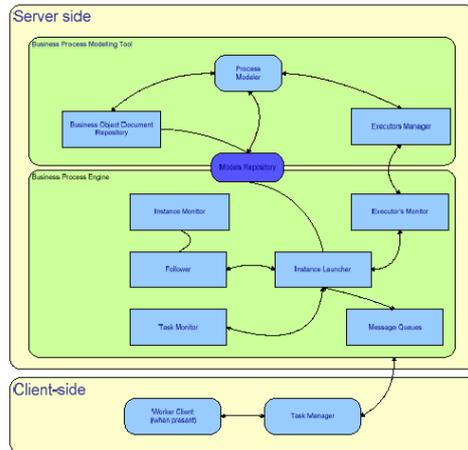


Ilustración 2. Arquitectura básica de la plataforma IDIERE (Franco et al , 2005)

El diseño de una plataforma que se propone debe asentarse sólidamente y formar parte de un concepto más amplio que es el de BPMS (Business Process Management System) del cual formaría parte importante. En la ilustración 2, se representa la arquitectura básica del BPMS en el cual se va a enmarcar el programador de actividades Franco et al. (2005). Como se puede observar, el BPMS propuesto se distribuye de forma que una parte de la gestión del proceso de negocio se realiza de forma centralizada, “Server Side”, y otra es realizada por cada recurso de forma autónoma, “Client Side”. Dentro de la propuesta de Franco et al. (2005) hay que destacar por su relación con la propuesta actual los componentes llamados “Task Monitor” y “Task Manager”, ya que son estos elementos los que son ahora diseñados como un Sistema Multi-Agente con el nombre de Plataforma de Programación de Actividades de Procesos de Negocio Extendido.

El Sistema Multiagente (MAS) que se propone está basado en la arquitectura MAS conocida como RETSINA de Sycara et al. (2003).

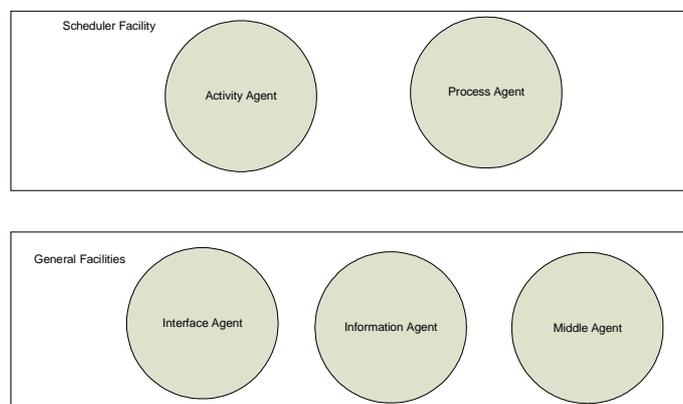


Ilustración 3. Agentes básicos del Sistema Multiagente Propuesto.

Retsina se basa en una arquitectura funcional consistente en cuatro agentes básicos.

- Interfase Agent, que actúa como enlace con los usuarios, envía y recibe resultados.
- Task Agent, que ayuda a los usuarios a realizar sus tareas y es el autentico núcleo de funcional del sistema en cuanto a sus objetivos.
- Information Agent, que proporciona a un conjunto heterogéneo de información, como son bases de datos.
- Middle Agent, que ayudan que los agentes puedan encontrar a otros agentes necesarios para alcanzar sus objetivos.

Dentro de PNE\_AGENT el Task Agent se ha denominado Scheduler Facility y está compuesto por 2 agentes software encargados de realizar la programación de las actividades de cada proceso de negocio que se reciba en el sistema. Por otro lado, se encuentran el Interfase Agent, Information Agent y Middle Agent a los que se ha denominado General Facilities, y que son los encargados de proporcionar la funcionalidad general y común en cualquier sistema multiagente.

Los agentes pertenecientes al Scheduler Facility son los encargados de recibir las peticiones de ejecución de un proceso, analizar su descomposición en actividades según su regla de negocio, y asignar los recursos adecuados para realizar el proceso intentando minimizar el objetivo establecido. Pero además de ejecutar este proceso de programación deben estar atentos a cualquier incidencia y actuar en consecuencia reprogramando la secuencia afectada. El agente Process Agent se encuentra físicamente en el lado del servidor, mientras que cada recurso está representado por un Activity Agent. La arquitectura de cada uno de estos agentes y su funcionalidad básica se describe a continuación.

El Process Agent tiene como objetivo fundamental ser capaz de recibir e interpretar cada una de las solicitudes de ejecución de un proceso de negocio (descomposición en actividades y reglas de precedencias), asignar o reasignar un recurso para realizar una actividad del proceso, e interpretar los eventos que recibe y actuar en consecuencia. La ilustración 4, muestra los bloques arquitectónicos del agente.

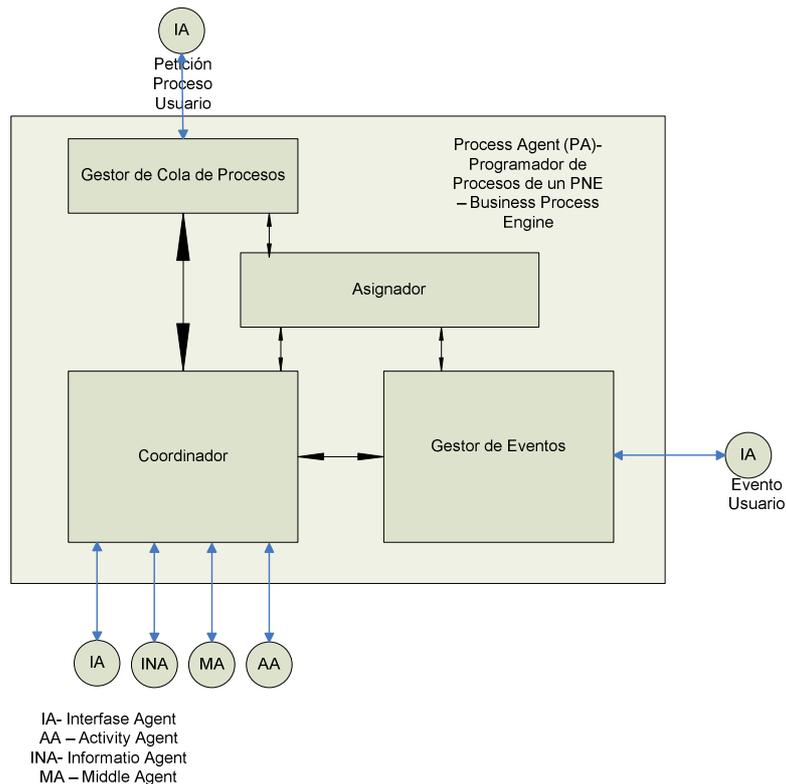


Ilustración 4. Arquitectura del Process Agent

*Gestor de Cola de Procesos:* Recibe las peticiones a través del Interfase Agent (IA) y los preprocesa con ayuda del bloque de coordinación. El preproceso básico consiste ordenar los eventos según una heurística para posteriormente ser descompuestos en actividades y obtener información adicional sobre el proceso (tiempos medios de ejecución, hora de llegada, reglas especiales de negocio, etc.)

*Asignador:* Es el bloque encargado de decidir que recurso de todos los disponibles para una determinada actividad es el asignado a ésta.

*Coordinador:* Es el encargado de asistir internamente al resto de bloques estableciendo si es necesario comunicación con el resto de agentes. Su función principal es que una vez descompuesto en actividades todos los procesos sigan un flujo de ejecución adecuado.

*Gestor de Eventos:* Es el encargado de recibir los eventos de los usuarios a través del Interfase Agent (cancelación un proceso, modificación prioridad, etc.). Los eventos que afectan a los recursos son atendidos por el Activity Agent.

El Activity Agent tiene como objetivo fundamental ser capaz de recibir las solicitudes de ejecución de una actividad y secuenciar la cola del recurso para realizar una actividad del proceso, así como interpretar los eventos que recibe y actuar en consecuencia. La ilustración 5, muestra los bloques arquitectónicos del agente.

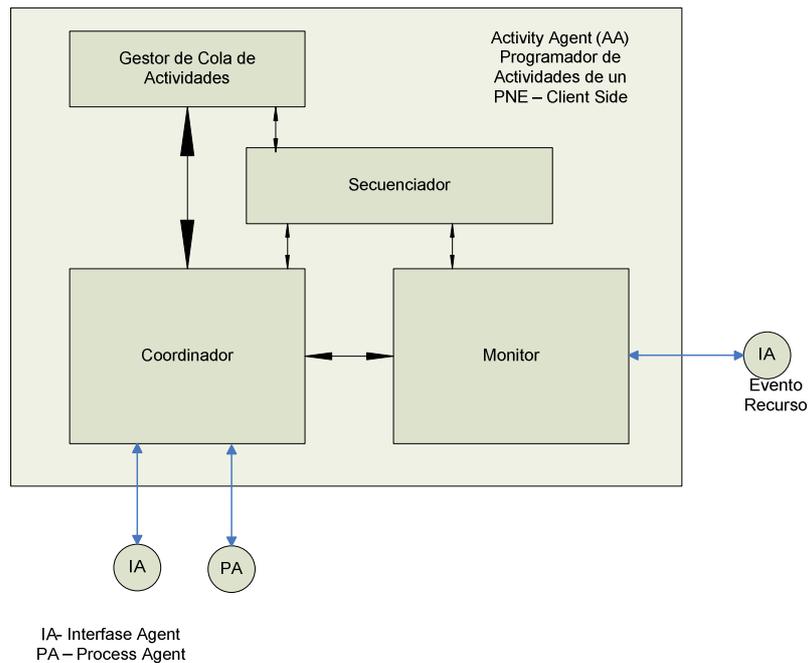


Ilustración 5. Arquitectura del Activity Agent

*Gestor de Cola de Actividades:* Recibe las peticiones a través del coordinador. Se realiza un preproceso simple consistente en obtener todos los datos necesarios para realizar la secuenciación.

*Secuenciador:* Es el bloque encargado de decidir el orden en que se ejecutan las actividades. Este bloque junto con el Asignador tiene embebido un algoritmo que tiene como objetivo encontrar los mejores resultados para la función objetivo.

*Coordinador:* Es el encargado de asistir internamente al resto de bloques estableciendo si es necesario comunicación con el resto de agentes.

*Gestor de Eventos:* Es el encargado de recibir los eventos del recurso a través del Interfase Agent (no disponibilidad, cambio de estado, cambio de calidad, etc..).

Una vez presenta la composición arquitectónica de la Plataforma Multiagente, es necesario especificar con mayor cual es el procedimiento de asignación-secuenciación que se implementa fundamentalmente en los bloques Asignador-Secuenciador. Dicho proceso se realiza a partir del algoritmo utilizado en Wu y Weng (2005). De tal forma que el protocolo que sigue el Asignador es:

1. Envía una solicitud de oferta al Activity Agent.
2. El Process Agent avalúa la oferta de los Activity Agent.
3. Selecciona el recurso.
4. Si todas las operaciones de un proceso han terminado finalizamos y sino volvemos al paso 1.

La asignación de cada actividad se realiza de forma secuencial, es decir, sólo se programa una actividad cuando la inmediatamente precedente se ha realizado. En todo caso, el Process Agent selecciona la oferta que menor valor aporta a la función objetivo, es decir con menor  $q(i,j,r) \cdot T_i$ .

Por otro lado el protocolo que se establece para el secuenciador es:

1. Recibe una solicitud de oferta desde el Process Agent.
2. Formula una oferta.
3. Envía la oferta al Process Agent.
4. Añade el trabajo a la cola del Activity Agent.
5. Procesa el trabajo según la secuencia.

Donde la formulación de la oferta es elaborada en base al proceso propuesto por Wu y Weng (2005) para el caso los trabajos TOLJ del Agent Machine.

El sistema propuesto no sólo garantiza la programación del conjunto de actividades sino que es capaz de realizar una reprogramación de las mismas en función de los eventos que ocurran a lo largo del tiempo.

Aunque no es considerado como un evento, si es importante, matizar que la propia dinámica del proceso es contemplada por el sistema. En el caso de procesos que tienen diversas alternativas que se van fijando en tiempo de ejecución (por ejemplo, que una factura se pague o se anule) el sistema propuesto actúa e forma similar al resto de los casos resultando siempre una programa factible. El Process Agent siempre consulta la base de datos de reglas, junto con el estado actual del proceso, para establecer cual es la siguiente actividad. De esta forma, procesos con alternativas y sin alternativas definidas en tiempo de ejecución son tratados de la misma forma.

## **5. La Propuesta del Sistema Multiagente**

El sistema multiagente que se propone tiene como objetivo afrontar la programación de actividades dentro de un sistema de gestión de negocio extendido de forma dinámica.

### **5.1. Descripción del Proceso de Modelado**

Los agentes son una potente herramienta de abstracción para el diseño y construcción de sistemas complejos, debido a que ofrecen una vía adecuada para considerar sistemas con múltiples componentes diferentes. En este trabajo se presenta una experiencia de modelado usando la metodología INGENIAS (Pavón y Gómez (2003)) para desarrollar una solución orientada a agentes para el problema de programación de actividades.

Durante las fases de análisis y diseño, se usan cinco meta-modelos: (i) meta-modelo organizacional, que define como se agrupan los agentes y cuales son las funcionalidades del sistema y las restricciones en el comportamiento de los agentes (ii) meta-modelo de agente, que describe los agentes concretos que se deben usar y sus estados mentales (iii) meta-modelo de interacción, que describe como los agentes interactúan y se coordinan entre ellos, (iv) meta-modelo de entorno, que define el tipo de recursos y aplicaciones a utilizar y (v) meta-modelo de tareas y objetivos, que relaciona los estados mentales de los agentes con cada tarea.

Este trabajo se centra en la fase de diseño del sistema de programación de procesos de negocio de un BPMS (ver ilustración 2). Se toma como base del diseño la propuesta realizada en el apartado 4, la cual se refina y en la que finalmente se asignan diversos

papeles en relación con los módulos vinculados en la programación del proceso de negocio. Es necesario incluir módulos como el Asignador, Secuenciador, Coordinador, etc. Con los que existe una interacción directa. El sistema puede ser fácilmente conectado con otros subsistemas de la del BPMS para implementar un sistema ágil de Gestión de Procesos de Negocio Extendido.

## **6. Conclusiones**

Este trabajo presenta un desarrollo metodológico, basado en la tecnología de multiagente, para el problema de Programación de Procesos de Negocio Extendido. Se ha presentado la necesidad real de disponer de una programación de procesos dinámicos y que permitan que la gestión empresarial pueda realizarse atendiendo a los eventos que se reciben on-line. La propuesta se basa en un proyecto a medio plazo. Su objetivo principal consiste en el desarrollar un sistema de gestión de procesos capaz de equilibrar la calidad en la ejecución de las actividades que componen un procesos junto con capacidad de cumplir los plazos establecidos. Esta propuesta se basa en la capacidad del sistema de ofrecer la alternativa más conveniente del producto aunque implique cambios en la programación, si se mejora la función objetivo global. Para satisfacer esa meta, es necesario integrar los diversos pasos distribuidos de la programación de una manera flexible, adaptable, versátil, robusta y natural. La tecnología de los sistemas de Agent/Multi-agent (MAS) se ha utilizado en la resolución de este problema, puesto que proporciona las características requeridas para los sistemas de gestión. Específicamente, hemos presentado una experiencia que modelaba usando la caja de herramientas de IDK de la metodología de INGENIAS, que se ha empleado con éxito en otros dominios.

## **Agradecimientos**

Este trabajo se ha desarrollado en el marco del proyecto DPI2004-02594- Interoperabilidad de Procesos Extendidos, que ha sido financiado con fondos FEDER y del Ministerio de Educación y Ciencia

## **Referencias**

- Franco, R. D.; Ortiz, A.; Lario, F. y Navarro R., 2001, Value Networks: Business Process Integration over the Internet, eBusiness. Key sigues, applications and technologies. IOS Press. ISBN 1-58603-205-4.
- Franco, R. D.; Anaya, V.; Ortiz, A., 2005, An Architecture for Managing Distributed Business Processes in Networked Organizations, I-ESA05 Interoperability of Enterprise software and Applications, ISBN 1-84628-151-2, Ginebra (Suiza)
- Galbraith, J. 1973. *Designing complex organizations*. Massachussets: Addison-Wesley Pub. Co.
- Katia Sycara, Joseph A. Giampapa, Brent K. Langley, and Massimo Paolucci "The RETSINA MAS, a Case Study," Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications, Alessandro Garcia, Carlos Lucena, Franco Zambonelli, Andrea Omici, Jaelson Castro, ed., Springer-Verlag, Berlin Heidelberg, Vol. LNCS 2603, July, 2003, pp. 232—250

Pavon, J.; Gomez, J. (2003). Agent Oriented Software Engineering with INGENIAS. Multi-Agent Systems and Applications II, Lecture Notes in Artificial Intelligence, 2691, 394–403.

Schneeweiss, C. 2003. *Distributed Decision Making*. Springer.

Vieira, G.E., Herrmann, J.W., and Lin, E. 2003. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1): 39-62.

Wu, Z. y Weng M., 2005, Multiagent Scheduling Method with Earliness and Tardiness Objectives in Flexible Job Shops, *IEEE Transactions on Systems, Man and Cybernetics, Part B. Cybernetics*, vol35, n° 2, April.