Gestión eficiente de carteras de proyectos.

Propuesta de un sistema inteligente de soporte a la decisión para oficinas técnicas y empresas consultoras.

Efficient project portfolio management. An intelligent decision support system for engineering and consultancy firms

• José Alberto Araúzo-Araúzo	Dr. Ingeniero Industrial	Escuela Técnica Superior de Ingenieros Industriales. Valladolid
• José Manuel Galán-Ordax	Dr. Ingeniero Industrial	Escuela Politécnica Superior. Burgos
• Javier Pajares-Gutiérrez	Dr. Ingeniero Industrial	Escuela Técnica Superior de Ingenieros Industriales. Valladolid
• Adolfo López-Paredes	Dr. Ingeniero Industrial	Escuela Técnica Superior de Ingenieros Industriales Valladolid

Recibido: 01/12/08 • Aceptado: 10/03/09

ABSTRACT

- In this paper, we show a "financial market inspired" approach to resource allocation in multiproject environments. This novel approach can give support to engineering and consultancy firms, engaged in project portfolio planning and scheduling in dynamic environments. Project importance and resources availability change over time because of changes in firm strategy, because of individual project overruns or simply, as a result of environment changes. Therefore, resources need to be re-allocated dynamically.
- In our methodology, projects and resources play the role of economic agents. Projects are defined by means of tasks to be performed according to defined precedence relations. Resources are endowed with skills and competences (knowledge, labour time, etc.) which are necessary to perform tasks. Projects ask for resources and resources offer their capabilities depending on their time slot availability. An "auction" takes place and the price of resources in a particular time-slot reflects the relation between availability and demand of resource capabilities.
- We describe an example to show how the methodology works. A pilot model can be used in Technical Engineering Offices in order to support portfolio scheduling decisions.
- **Key words:** project portfolio management, Project scheduling, multiproject environments, multiagent systems, auction.

RESUMEN

En este artículo demostramos cómo una aproximación inspirada en los procedimientos de asignación utilizados en los mercados financieros permite una asignación eficiente de recursos a proyectos en un entorno multiproyecto. Presentamos una herramienta desarrollada para dar soporte en tiempo real a las necesidades de planificación y programación de la cartera de proyectos de una empresa de ingeniería o una empresa consultora. Todo ello en un entorno dinámico donde la cartera de proyectos, sus propiedades, así como el fondo de recursos y sus características cambian, bien por evolución de la estrategia de la compañía, por retrasos o sobre costes en alguno de los proyectos individuales, o simplemente por modificaciones en el entorno.

En el sistema propuesto, proyectos y recursos emulan a agentes económicos. Los proyectos, que son descritos mediante sus tareas y relaciones de precedencia, buscan recursos para intentar finalizar en el plazo determinado. Los recursos están dotados de ciertas habilidades o competencias (conocimientos, fuerza de trabajo, etc.) para ejecutar alguna de esas tareas. En este modelo, los proyectos demandan parte del tiempo disponible a los recursos que disponen de las capacidades para ejecutar sus tareas. El precio establecido mediante una "subasta" refleja la relación entre la disponibilidad y la demanda de las competencias asociadas al recurso.

Para demostrar la utilidad, se plantea un problema en el que mostramos cómo opera el sistema desarrollado. Este modelo piloto puede aplicarse en oficinas técnicas de ingeniería y empresas consultoras fácilmente, ayudando de forma integrada

Los gestores deben priorizar y seleccionar los proyectos (de entre los propuestos por los clientes) que se añadirán a la cartera, teniendo en cuenta los márgenes esperados y el estado de saturación de los puestos de trabajo (recursos).

a la toma de decisiónes sobre la composición y la programación de carteras de proyectos.

Palabras clave: gestión de la cartera de proyectos, programación de proyectos, entornos multiproyecto, sistemas multiagente, subasta.

1.- GESTIÓN DE CARTERAS DE PROYECTOS

La literatura sobre dirección de proyectos se ha centrado fundamentalmente en la gestión de proyectos individuales. Pero en la práctica, las empresas trabajan generalmente en entornos multiproyecto, donde varios proyectos se están ejecutando simultáneamente usando para ello un conjunto limitado de recursos.

Un ejemplo de este tipo de entornos es una oficina de proyectos de ingeniería (Martínez et al., 2001) o una empresa consultora, donde trabajan personas dotadas de diferentes habilidades que realizan ciertos tipos de tareas (cálculos, redacción, realización de planos, revisión, etc.). En una oficina de este tipo, los nuevos proyectos se integran en la cartera de la empresa a medida que el cliente los va demandando. Cada proyecto exige la realización de un conjunto de tareas, ejecutadas en un orden concreto y que sólo pueden ser llevadas a cabo por personas que posean las habilidades necesarias.

En estos entornos, los gestores deben priorizar y seleccionar los proyectos (de entre los propuestos por los clientes) que se añadirán a la cartera, teniendo en cuenta los márgenes esperados y el estado de saturación de los puestos de trabajo (recursos). Además, deben repartir las tareas entre los diferentes trabajadores, priorizar los trabajos y programar su ejecución. Todo ello, buscando el logro de los objetivos operativos y estratégicos de la empresa

Debido a la adición de nuevos proyectos, a cambios en la estrategia corporativa, o simplemente a la información obtenida de la retroalimentación del sistema, la lista de prioridades cambia a lo largo del tiempo. Todos estos cambios unidos a las perturbaciones que suceden en los recursos (fallos en equipos, retrasos en la ejecución de los trabajos, etc.) obligan a la reasignación y reprogramación de tareas, buscando de nuevo la mejor forma de conseguir los objetivos de la empresa.

Las perturbaciones y los cambios en la cartera de proyecto pueden suponer nuevos conflictos entre proyectos (p.e., varios proyectos compiten por el mismo periodo de tiempo de un mismo recurso). Por este motivo la dirección eficiente de carteras de proyectos requiere del desarrollo de sistemas inteligentes de soporte a la decisión, que faciliten la toma de decisiones en cuanto a asignación de recursos y programación de tareas, siempre teniendo en cuenta la limitación de recursos, y el cumplimiento del plazo de los proyectos de la cartera.

Desafortunadamente, la investigación sobre la asignación de recursos en entornos multiproyecto no ha proporcionado una solución suficientemente robusta y flexible para los casos más generales (Anavi-Isakow & Golany, 2003). En Hans et al (2007) se puede encontrar una revisión de la literatura existente, centrándose en modelos de asignación jerárquicos (primero se asignan recursos a cada proyecto según su importancia estratégica, y posteriormente, cada proyectos gestiona independientemente los recursos que le han sido asignados). Además, propone una plataforma genérica de programación y control de proyectos que basada en la selección de diferentes métodos en función de cuestiones organizacionales. En Kao et al (2006) se propone un sistema dirigido por eventos para desarrollar una plataforma de decisión trade-off para la programación y reprogramación. Cohen et al. (2004) analizan el comportamiento del método de la cadena crítica de Goldratt (1997) en entornos multiproyecto.

En este artículo proponemos un sistema inteligente de soporte a la decisión (IDSS) que opera en tiempo real. Se trata de una nueva aproximación para la programación dinámica de tareas en tiempo real en entornos multiproyecto ejecutados por una organización dada. Con esta herramienta seremos capaces de analizar algunos de los problemas más difíciles relacionados con la gestión de la cartera de proyectos: la determinación de los recursos ociosos y sobrecargados; la detección de los recursos clave

y la de las habilidades que deben incrementarse en el sistema; la conveniencia de incorporar nuevos recursos; el análisis del papel de la flexibilidad de los recursos, o lo que es lo mismo, la posibilidad de desempeño de varias habilidades por un mismo recurso; el efecto de adición o eliminación de proyectos; el cambio de prioridades en tiempo de ejecución; etc.

El resto del artículo se organiza como sigue: primero presentamos la metodología que soporta el IDSS que hemos implementado. A continuación definimos formalmente el problema que estudiamos: la programación de la cartera de proyectos con posibilidad de rechazo, y con limitación de recursos. En las siguientes secciones desarrollamos el modelo multiagente subyacente, y los resultados obtenidos en un periodo de simulación. Finalmente presentamos las conclusiones y extensiones de nuestro trabajo.

2.- SISTEMAS MULTIAGENTE Y ENTORNOS MULTIPROYECTO.

Los proyectos se caracterizan por su complejidad estática (existen muchos elementos y dependencias), por su incertidumbre (sobre la disponibilidad de recursos, duración de las tareas), por el comportamiento dinámico (cambios en el alcance del proyecto, incorporación de tareas, reprogramación) y por su natural distribución (cada operación puede ser ejecutada por diferentes recursos y en diferentes localizaciones geográficas).

Todas estas características se ven intensificadas en el caso de entornos multiproyecto. Mas proyectos implica más complejidad estática, más incertidumbre, más complejidad dinámica, y más dispersión geográfica ya que no sólo las actividades y los recursos estarán distribuidos, también pueden estarlo la dirección de cada proyecto.

Dentro de la inteligencia artificial distribuida existe un paradigma denominado sistema multiagente, que resulta muy útil para analizar y diseñar nuestro sistema. Los sistemas multiagente se han concebido precisamente para abordar problemas con características similares al nuestro. Estos sistemas son adecuados para tratar problemas que se caracterizan por ser complejos, abiertos (los componentes que aparecerán en el sistema no se conocen con antelación, pueden cambiar a lo largo del tiempo, son altamente heterogéneos y dinámicos), dinámicos, estocásticos y ubicuos (la actividad y la toma de decisión se distribuye a lo largo de toda la estructura del sistema) (Jennings & Wooldridge, 1995; Pavón & Pérez, 2005; Wooldridge, 2002).

Un sistema multiagente está formado por ciertas entidades software a las que dotamos de capacidad de interaccionar entre ellas y con un entorno. Como objetos[1] que son, poseen un estado (datos, conocimiento); pero además son autónomos (no sólo actúan en base a instrucciones, poseen además reglas para la toma de decisión autónoma); reactivos (reaccionan ante estímulos externos); proactivos (tienen objetivos propios y realizan sus propios planes para conseguirlos); y sociales (interaccionan con otros agentes). Esta forma tan directa de abstraer los sistemas ha favorecido el desarrollo de sistemas basados en agentes en múltiples campos (véase por ejemplo, Pajares et al. (2004), López et al. (2005), Pascual et al. (2006), Galán et al. (2008), Posada & López (2008))

En el caso particular de los entornos multiproyecto, las entidades que son susceptibles de ser abstraídas como agentes pueden ser las tareas, los recursos, los proyectos, etc. Este diseño permite distribuir la gestión del sistema en componentes elementales directamente identificables en la organización que es objeto de estudio. De esta forma se facilita el diseño del sistema de información, así como su mantenimiento y su posible adaptación a escenarios no previstos inicialmente. Dado que el sistema de información se distribuye de acuerdo a la estructura física u organizacional del entorno, cualquier cambio en esta estructura es fácilmente trasladable al sistema de gestión (Araúzo et al., 2008).

Estas aproximaciones descentralizadas se han propuesto para la gestión de proyectos desde la década pasada (Yan et al., 2000), pero es en los últimos años cuando están recibiendo un creciente interés, especialmente aquellas que se basan en analogías con los mercados competitivos (Clearwater, 1996). Recientemente, Lee, Kumara y Chatterjee (Kumara et al., 2002; Lee et al., 2003) han propuesto un sistema multiagente para la programación dinámica de tareas en un entorno distribuido de múltiples proyectos utilizando para ello mecanismos de mercado. En la misma línea se encuentra el trabajo de Confessore et al (2007), donde se describe un mecanismo iterativo de subasta combinatoria que resuelve el mismo problema.

Otros ejemplos de aproximaciones basadas en agentes en el campo de la dirección de proyectos se pueden encontrar en las publicaciones de Kim et al. (Kim et al., 2000; Kim & Paulson, 2003b; Kim & Paulson, 2003a; Kim et al., 2003) de Wu and Kotak (2003) o de Cabac (2007). Nuestro trabajo continua con esta línea de investigación, centrándose concretamente en el uso de la metáfora de mercado y de mecanismos de mercados para el desarrollo de sistemas multiagente que ayuden a los gestores de proyectos a la toma de decisión sobre el uso de los recursos, y sobre la composición de la cartera. Este trabajo añade otro paso en el uso de las herramientas proporcionadas por los sistemas multiagente como sistemas de soporte para la decisión en entornos multiproyecto.

1 En sentido computacional

3.- PROGRAMACIÓN MULTIPROYECTO CON POSIBILIDAD DE RECHAZO.

La formalización del problema de programación multiproyecto con posibilidad de rechazo de proyectos que abordamos en este trabajo es el siguiente:

En cualquier instante t existen I proyectos en el sistema, cada uno de ellos denotado mediante i (Figura 1). Cada proyecto se caracteriza por un valor V_i, que puede interpretarse como el pago que un cliente hace por el proyecto o como la valoración que un proyecto interno tiene para la organización, un peso w_i que representa la importancia estratégica del proyecto, una fecha de entrega deseable D_i, un fecha de entrega límite D_i^* que no puede ser excedida, una fecha de llegada del proyecto al sistema B_i, y una fecha de respuesta límite R_i que es la fecha mas tardía para decidir si el proyecto es rechazado o incluido en nuestra cartera. El sistema es dinámico: los proyectos llegan al sistema o son rechazados mientras otros mientras otros proyectos se están ejecutando.

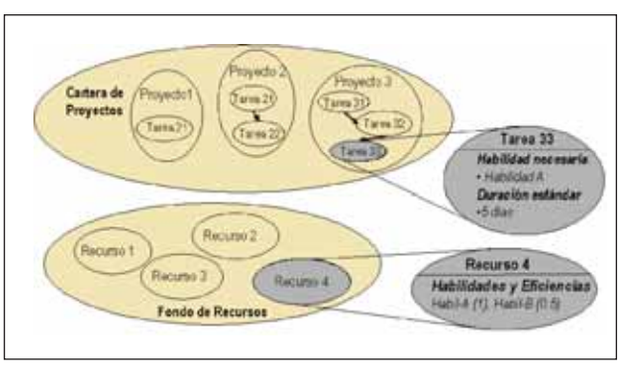

Figura 1. Entorno Multiproyecto.

Cada proyecto i consiste en J_i actividades o tareas, cada una de ellas denotada por los subíndices ij donde i{1, 2,…, I} y j{1, 2,…, J_i}. Cada tarea lleva asociada una duración estándar d_{ij} (duración de la actividad en un recurso de eficiencia normal).

La empresa gestiona un conjunto M de recursos encargados de completar las actividades de los proyectos. Cada recurso m {1, 2, 3…M} sólo podrá ejecutar simultáneamente una única tarea, que una vez comenzada deberá ser finalizada.

Para llevar a cabo los proyectos pueden ser necesarias un conjunto H={h_1, h_2,...,h_K} de habilidades por parte de los recursos. Cada recurso está dotado de un ratio de coste por unidad de tiempo C_m y de un subconjunto H_m de habilidades que es capaz de desarrollar. Los recursos tienen diferente grado de pericia o destreza para realizar cada habilidad. Así, la capacidad de trabajo de los recursos se puede representar mediante un vector de destrezas en habilidades del recurso m e_m=(e_{m1}, e_{m2},…,e_K) donde $e_{mf} \geq 0$ y mide la destreza del recurso m para realizar la habilidad f. Si $e_{mf} = 0$ entonces el recurso no poseerá la habilidad k_f, si $0<e_{mf}<1$ el recurso exhibirá la habilidad k_f ineficientemente, si $e_{mf} = 1$ lo hará con eficiencia estándar, y si $e_{mf} > 1$ su eficiencia será elevada.

> Debido a la adición de nuevos proyectos, a cambios en la estrategia corporativa, o simplemente a la información obtenida de la retroalimentación del sistema, la lista de prioridades cambia a lo largo del tiempo.

Cada actividad está asociada a una habilidad k(ij) H. Una actividad ij asociada a una habilidad k(ij) puede ser realizada por el recurso m si y sólo si $e_{m,k(ij)}>0$, es decir, si el recurso posee una destreza no nula en la habilidad necesaria para desarrollar la actividad ij.

La duración de la tarea ij asignada a un recurso m (d_{ijm}) se calculará en función de la duración estándar de la actividad (d_{ij}) y de la eficiencia con la que el recurso desempeña la habilidad necesaria k(ij), por lo que la duración de la actividad dependerá del recurso al que se la asigna según la expresión $d_{ijm}=d_{ij}/e_{m,k(ij)}$.

En nuestro primer modelo asumiremos, para facilitar la interpretación de los resultados, que dentro de cada uno de los proyectos sus actividades se deben ejecutar secuencialmente, una tras otra, según el orden indicado por el subíndice j. La eliminación de este supuesto no supondría añadir problemas conceptuales al modelo, pero aumentaría la complejidad de alguno de los algoritmos incorporados. En todo caso, esto no impide extrapolar los resultados obtenidos a casos más complejos.

También supondremos que las tareas, una vez comenzadas, no se podrán interrumpir. El recurso trabajará sobre la actividad asignada hasta que la finalice, después podrá pasar a ejecutar otra tarea.

Se incluye la posibilidad de reasignación de recursos en tiempo real cada vez que un nuevo proyecto llega al sistema o cuando los cambios del entorno lo justifiquen (pérdida o incorporación de un recurso, anulación de un proyecto, etc.).

La eficiencia global del sistema se evaluará mediante el beneficio medio obtenido por unidad de tiempo. Se calculará de acuerdo a la siguiente expresión, para un horizonte temporal T.

$$Efficiency = \frac{B_T}{T} = \frac{\sum_i (V_i - Cost(i))}{T}$$

Expresión 1

donde i indica cada uno de los proyectos finalizados durante T, y Cost(i) es el coste asociado al proyecto. Este coste tiene dos componentes, el coste directo de uso de los recursos y un coste asociado al retraso.

$$Cost_i = \sum_{j=1}^{J_i} c_{m(i,j)} \cdot d_{ijm} + w_i \cdot \phi$$

$$con \quad \phi = \begin{cases} 0 & si\ F_i \le D_i \\ (F_i - D_i)^2 & si\ D_i < F_i \le D_i^* \\ \infty & si\ F_i > D_i^* \end{cases}$$

Expresión 2

El primer término se corresponde con el coste del tiempo de uso los recursos para ejecutar cada tarea ij. $c_{m(i,j)}$ denota el coste por unidad de tiempo del recurso m(i,j), donde m(i,j) indica el recurso seleccionado para realizar la operación j del proyecto i. El segundo término es el coste de retraso asociado al proyecto, donde F_i es la fecha real de finalización del proyecto. Este coste será cero si el proyecto finaliza antes de la fecha deseada D_i, si el proyecto finaliza entre la fecha deseada y la fecha límite el coste será proporcional al cuadrado del retraso, y si la fecha de finalización supera la fecha limite D_i^* el coste tenderá a infinito. La consideración de los costes por retraso de forma cuadrática implica que los altos retrasos son mucho más costosos que los retrasos moderados. De esta de forma nuestro sistema va a penalizará altamente los retrasos elevados, optando por soluciones sin retrasos extremos. Además esta forma de proceder garantiza una mejor convergencia del proceso de subasta.

El retraso sobre el plazo límite, se penaliza con un coste infinito para no considerar durante el proceso de programación y selección de proyectos, proyectos con programas que violen este plazo. Un vez que el proyecto ha sido aceptado, el sistema hará todo lo posible para que no se supere el plazo límite, pero puede haber algunos casos en los que esto no sea factible (retrasos en la ejecución de operaciones, absentismo,…). En esos casos se finalizará tras la fecha límite y se computarán los retrasos y sus costes a partir de la fecha de finalización deseada.

Se considera la posibilidad de rechazar proyectos, lo que puede suceder por los siguientes motivos.

- El valor aportado por el proyecto no compensa los costes.
- El proyecto no se puede finalizar antes de su fecha límite D_i^*.
- El impacto sobre la programación del resto de proyectos es inaceptable. Esto puede suceder por ejemplo, cuando la inclusión del nuevo proyecto obligaría a retrasar los proyectos comprometidos hasta superar su fecha D_c^*, o cuando la inclusión de un nuevo proyecto incrementaría los costes de retraso de los otros proyectos más que el margen obtenido por la inclusión del nuevo.

La cartera puede estar compuesta por proyectos de diversa naturaleza (aquellos realizados para clientes, proyectos de reorganización, de I+D, etc.). Puesto que los criterios de decisión utilizados son fundamentalmente económicos, para que el sistema pueda decidir sobre la ejecución de proyectos internos, estos se deberían valorar en términos monetarios. Si además algún proyecto tiene ventajas añadidas para la organización, como la captación de un cliente, etc, éstas deberían valorarse en términos económicos (coste de oportunidad que se está dispuesto a asumir por ese concepto). Si los proyectos y sus ventajas adicionales son de difícil valoración, nuestro sistema tiene posibilidad de incluirlos sin valorar y forzar al sistema su aceptación.

Inicialmente, no se contempla la posibilidad de subcontratar tareas, aunque el sistema puede extenderse para la toma de decisiones sobre subcontratación de tareas, así como para la selección entre posibles subcontratistas, con sólo conocer los costes asociados a la subcontratación, la posible duración de las tareas subcontratadas y la disponibilidad temporal del subcontratista para realizarlas. Tampoco se contempla la posibilidad de establecer plazos para hitos intermedios como pueden ser los “entregables”. No sería muy complicado eliminar este supuesto, ya que solo conlleva la introducción nuevas restricciones temporales que afectan a la programación local de los proyectos. En el estado inicial del prototipo no se han introducido estas funcionalidades por simplicidad, y para facilitar el análisis y la verificación de la robustez de la aproximación propuesta”.

4.- UN SISTEMA MULTIAGENTE PARA LA GESTIÓN ONLINE DE UNA CARTERA DE PROYECTOS.

Tradicionalmente los problemas de programación se han resuelto off-line mediante un sistema centralizado de toma de decisión, y usando para ello, un modelo de optimización global. Al contrario de lo que sucede con este paradigma tradicional, en nuestro sistema existirán varias entidades independientes capaces de tomar decisiones; entidades que serán modeladas como agentes. Consideraremos dos tipos de agentes, agentes Proyecto y agentes Recurso (Figura 2).

En un determinado momento el sistema poseerá tantos agentes Proyecto como proyectos se hayan propuesto. Cada uno de ellos representa a un proyecto concreto y poseerá información sobre sus operaciones, relaciones de precedencia, habilidades necesarias, valor, fechas de

entrega y estado de ejecución (grado de ejecución de las tareas planificadas). El objetivo de estos agentes será completar el proyecto, y para conseguirlo, acordarán con los recursos planes de actuación (programa local del proyecto), supervisarán el cumplimiento de estos planes, y los modificarán cuando la situación lo exija.

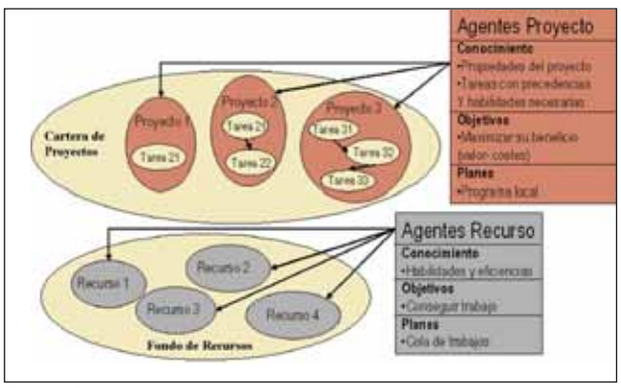

Figura 2. Sistema multiagente propuesto.

En el sistema existirán también tantos agentes Recurso como recursos sean considerados. Estos poseerán información sobre sus competencias, disponibilidad, coste temporal y sobre su estado (grado de ejecución de las tareas pendientes). Los agentes Recurso tendrán como objetivo conseguir trabajo y ocupar así su tiempo disponible. Este trabajo lo obtendrán de los proyectos, a través de acuerdos que se formalizarán mediante contratos, que conformarán la lista de tareas del recurso o programa local del recurso.

En nuestro sistema multiagente el sistema de toma de decisión es ubicuo, está distribuido entre todos los agentes, es decir, cada agente crea su propio programa local teniendo en cuenta para ello sus propios objetivos y su propia información. Esta forma de proceder puede originar programas locales incompatibles (varios proyectos programan el uso de un mismo recurso durante el mismo periodo), e ineficientes a nivel global (se desestiman los proyectos más rentables, se producen retrasos en los proyectos más importantes, etc.). Resulta por tanto necesario establecer un mecanismo de interacción entre agentes que minimice estos problemas. Este mecanismo de interacción se basa en una metáfora de mercado.

Para establecer la metáfora comenzaremos por subdividir los periodos de disponibilidad de los recursos en pequeños intervalos de tiempo o unidades temporales que denominaremos "slots de tiempo" (Figura 3). Cada "slot temporal" de cada recurso se modelará como un bien que el recurso desea vender y los proyectos desean comprar

Con esta analogía, un programa local será una asignación de un conjunto de "slots de tiempo" a un proyecto. Dado que en los mercados competitivos, se produce bajo ciertas condiciones una asignación óptima de recursos, se propone utilizar los mecanismos observados

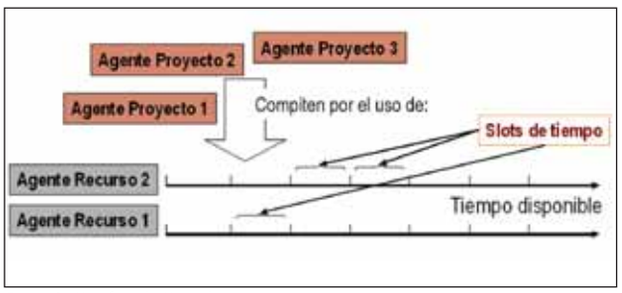

Figura 3. Metáfora de Mercado.

en estos mercados (subastas) para diseñar las interacciones entre agentes Proyecto y agentes Recurso, de forma que la asignación de slots a proyectos se realice de la forma más eficiente posible. Cada agente Proyecto actuará como un comprador en un mercado donde adquirir tiempo (slots) de los agentes Recurso, y donde estos intentarán vender su tiempo al máximo precio posible. Cada agente Proyecto, optará por un conjunto de slots que le permitan conseguir sus objetivos al mínimo coste. Es decir, cada agente Proyecto seleccionará un paquete de slots que le permitan ejecutar sus operaciones pendientes y además que minimice el coste de la expresión 3.

$$TC_i = \sum_{mt \in Z_i} p_{mt} + w_i \cdot \phi_i$$

TC_i : coste total del proyecto i.
p_{mt} : precio del slot temporal (t) del recurso (m)
Z_i : paquete de slots de tiempo seleccionados por el proyecto (i)

Expresión 3

Para seleccionar el paquete de slots adecuado, los agentes Proyecto utilizan un algoritmo de programación dinámica que evalúa el coste de todas las posibles combinaciones de slots que son adecuadas para ejecutar sus operaciones (Wang et al., 1997). Si algún proyecto no puede encontrar un conjunto de slots con los que pueda ejecutar el proyecto antes de D_i^*, con un coste menor a su valor (V_i), entonces no seleccionará ningún conjunto de slots. Esto significará que el proyecto no es rentable en la correspondiente iteración del proceso de subasta.

Cada agente Recurso determina el precio de sus slots temporales con el propósito de reducir los conflictos en el uso de recursos y maximizar su margen. Para ello se sigue un procedimiento basado en el algoritmo de optimización del subgradiente (Zhao et al., 1999), que en términos de subasta se puede interpretar como sigue. Cuando comienza la subasta el Recurso valora todos sus slots temporales a un precio mínimo que no se puede reducir (coste mínimo) que

será igual al coste del slot temporal. Después, cuando cada agente Proyecto ha seleccionado el paquete de slots temporales adecuado (según el procedimiento descrito anteriormente), los recursos incrementarán el precio de los slots donde existe conflicto (más de un proyecto ha seleccionado el slot temporal) y reducirán los precios (nunca por debajo del coste mínimo) de los slots temporales no demandados. Una vez se hayan ajustado los precios por parte de los recursos, los proyectos volverán a realizar sus propios programas locales (selección de slots temporales más satisfactorios). Tras ello, los recursos volverán a ajustar los precios, y así iterativamente. El proceso de subasta continúa indefinidamente.

Al comienzo de la subasta el precio de cada slot temporal será igual a t c_m (tamaño del slot temporal por el coste temporal asociado al recurso). En el resto de la subasta los precios de los slots temporales se ajustarán según la expresión 4.

$$p_{mt}^{\,n+1} = \max\left\{\Delta t \cdot c_m\,,\; p_{mt}^{\,n} + \alpha^n \cdot g_{mt}^n\right\}$$

Donde: $p_{mt}^{\,n+1}$: precio del slot (t) del recurso (m) en la iteración (n+1)

$p_{mt}^{\,n}$: precio del slot (t) del recurso (m) en la iteración (n)

α^n : paso en la iteración (n). Decrece cuando los precios convergen.

$g_{mt}^n = a_{mt}^n - 1$: subgradiente, donde a_{mt}^n es la demanda del slot (t) de (m)

Expresión 4

Mediante el mecanismo de subasta descrito anteriormente, los agentes Proyecto elaboran programas compatibles y globalmente eficientes. Además, al mismo tiempo, los agentes interactúan a través de un proceso complementario con el que realizan contratos en firme; contratos que estarán basados en los programas locales realizados en el proceso de subasta. Esos contratos determinan programas en firme para las tareas más inmediatas. Cuando esos contratos han sido realizados, los agentes Proyecto no considerarán nunca más las operaciones asociadas en los procesos de subasta.

Dependiendo del grado de ajuste de los precios a las condiciones reales del sistema, los programas locales serán más o menos compatibles y globalmente eficientes. Si en el momento de realizar los contratos en firme existen incompatibilidades, éstas se eliminarán usando para ello reglas heurísticas. Aunque esto no garantice la eficiencia global, sí que asegurará la compatibilidad de los programas.

En las Figuras 4 y 5 se muestran los diagramas de flujo que representan el funcionamiento básico de cada tipo de agente. Obsérvese que cada agente posee dos flujos básicos que se ejecutan en paralelo; mientras el flujo de la derecha representa las actividades relacionadas con la programación, el de la izquierda describe las tareas de ejecución y control.

En los agentes Proyecto (Figura 4), el flujo de programación esta destinado a la elaboración de programas localmente óptimos para unos precios determinados de los slots temporales de los recursos. El agente está a la espera de precios, y cada vez que recibe nuevos precios para los slots de los recursos, calcula un nuevo programa local que serán almacenados en 'Programa'. Se considerará que un tipo de programa local es descartar el proyecto.

En nuestro sistema multiagente el sistema de toma de decisión es ubicuo, está distribuido entre todos los agentes, es decir, cada agente crea su propio programa local teniendo en cuenta para ello sus propios objetivos y su propia información.

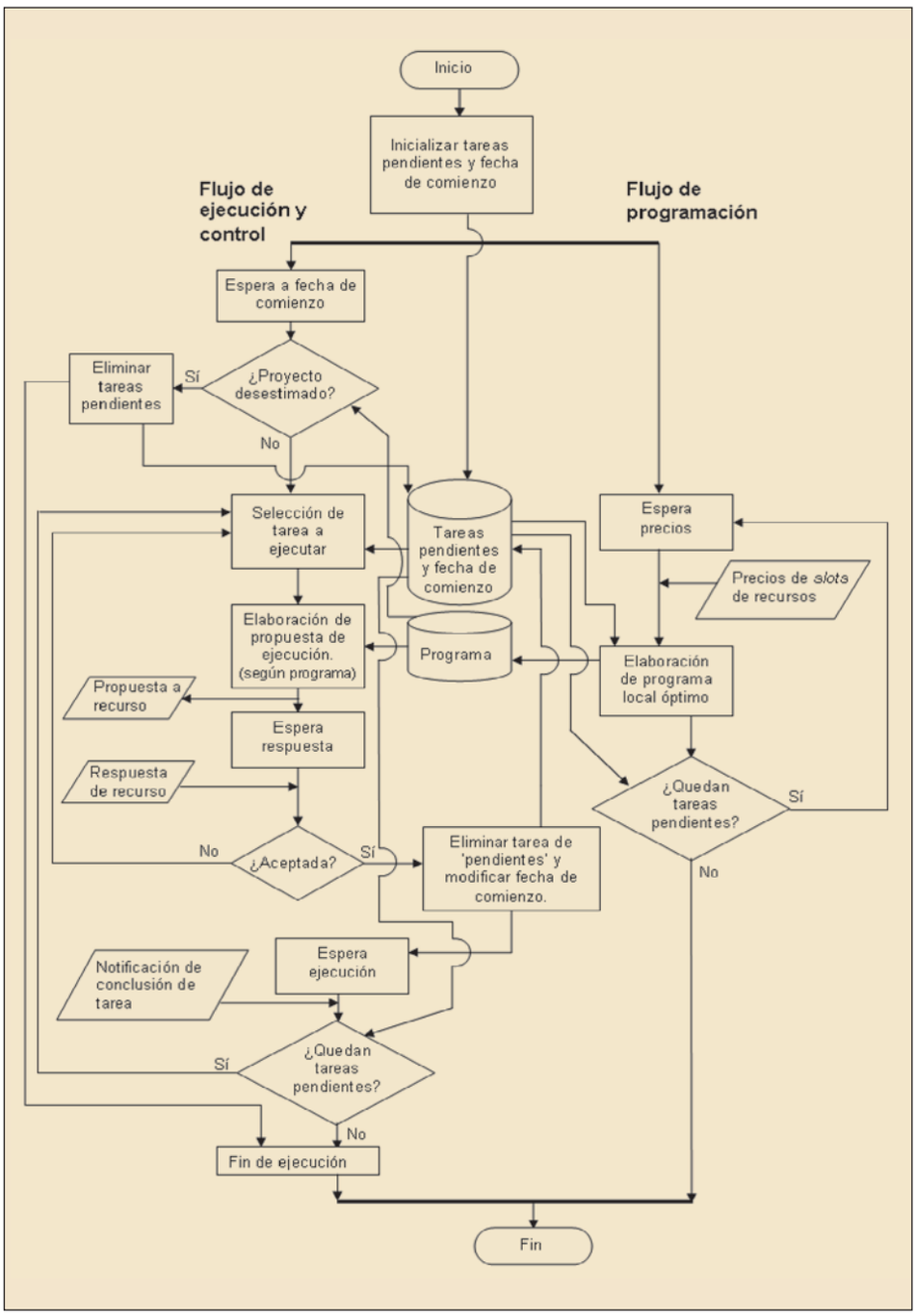

Figura 4. Diagrama de flujo de los agentes Proyecto.

De forma paralela a este proceso de elaboración de programas, los agentes Proyecto gestionarán la ejecución de las actividades, y ello, según los programas elaborados. Al crearse el agente y tras haberse inicializado la base de datos de tareas pendiente y fecha de comienzo, el agente comenzará esperando a la fecha de comienzo del proyecto. Cuando llegue este momento, el agente Proyecto, comprobará el programa local más actual elaborado. Si este programa sugiere la desestimación del proyecto, se eliminarán las tareas pendientes de la base de datos y se elimina el agente del sistema. Si por el contrario el programa no sugiere desestimar el proyecto, el agente continuará proponiendo a un recurso la ejecución de la tarea indicada en el programa local entre las fechas señaladas por dicho programa.

Tras la elaboración de la propuesta, los agentes Proyecto se la envían al agente Recurso seleccionado. Si el agente Recurso desestima la propuesta, se realizará una nueva. Si se acepta, se habrá producido un contrato en firme, y entonces el agente quedará a la espera de la notificación de su ejecución. Además, el agente elimina la tarea del conjunto de tareas pendientes y modifica la fecha de comienzo para los nuevos programas locales. Cuando la tarea se haya ejecutado, el agente Proyecto recibirá una notificación por parte del recurso, entonces comprobará si quedan tareas pendientes; si es así se volverá a realizar propuestas para estas tareas, si no, la ejecución del proyecto habrá finalizado y el agente se eliminará.

En la Figura 5 se tiene el flujo de control de los agentes recurso. En este caso el flujo de programación está dedicado al establecimiento y actualización de los precios de los slots temporales del recurso. Ello lo hace en función de los programas locales recibidos de los agentes proyecto siguiendo el procedimiento explicado con anterioridad. El flujo de programación y control hace referencia a la aceptación y rechazo de propuestas de trabajo recibidas desde los agentes proyecto, así como a la ejecución de estos trabajos.

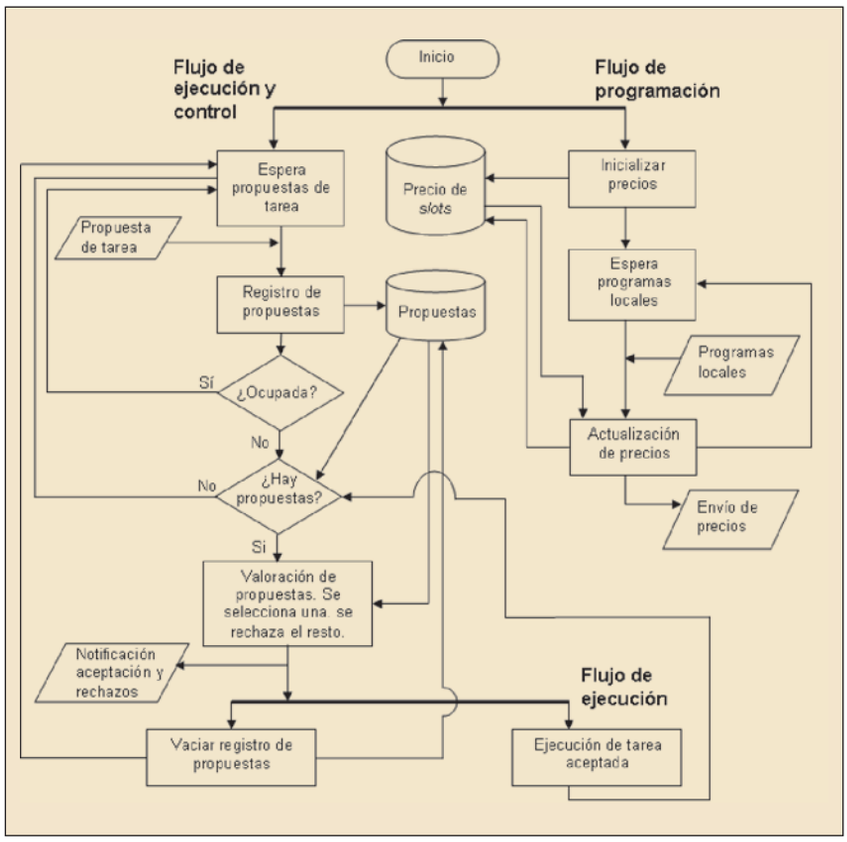

Figura 5. Diagrama de flujo de los agentes Recurso.

5.- CASO DE ESTUDIO

En esta sección mostraremos algunas simulaciones que ilustran cómo funciona el IDSS desarrollado, centrándonos en el manejo de la flexibilidad, es decir en el efecto del reparto de las habilidades entre los recursos del entorno multiproyecto.

Consideraremos tres recursos (R1, R2 y R3), que poseen respectivamente las habilidades C1, C2 y C3. En la tabla 1 se muestra una cartera de seis proyectos con sus operaciones. Cada tarea es caracterizada por la habilidad que es necesaria para ejecutarla así como por su duración estándar. Cada proyecto tiene además: una fecha de llagada (la fecha en la que el proyecto se incluye en el sistema, una fecha de comienzo, de las actividades (si el proyecto se desestima debe ser antes de esta fecha), una fecha de entrega deseable (D_i) y una fecha de entrega límite (D_i^*).

En la Figura 6 se muestra la evolución del sistema (los ejes horizontales representan el paso del tiempo) y la evolución de Gap de Dualidad (gráfica superior de la Figura 6). El gap de dualidad es una medida de la convergencia del proceso de subasta (Luh, Hoitomt, 1993). En los momentos de tiempo donde el gap de dualidad es bajo, el proceso de subasta ha convergido a unos precios de cuasi-equilibrio y por lo tanto los programas locales realizados bajo estas condiciones son bastante compatibles y globalmente eficientes. Las gráficas inferiores muestran, para cada recurso un diagrama de Gantt de los trabajos realizados por el recurso, y además, sobre el gráfico de Gantt, los precios de los slots temporales a los que se ha realizado la contratación.

Obsérvese que cuando los primeros proyectos son incluidos en el sistema, el gap de dualidad es elevado, pero entonces, a medida que el proceso de subasta se ejecuta, los precios se estabilizan y el gap de dualidad comienza a disminuir. Pero al incorporar nuevos proyectos al sistema en el instante de tiempo 30, el gap de dualidad aumenta. Esto es debido a que los precios en ese momento no son representativos del nuevo estado del sistema (nuevos proyectos se han incorporado). Tendrá que pasar un tiempo hasta que los precios se vuelvan a ajustar a las nuevas condiciones mediante el mecanismo de mercado. Después, cuando los precios se han desplazado hasta un estado de cuasi-equilibrio el gap de dualidad disminuye de nuevo.

Project	Tasks			Arrival date	Starting Date	D_i	D_i^*	Value
	Task 1	Task 2	Task 3					
P1	C1 50	C2 10	C3 20	-30	0	120	180	2500
P2	C3 10	C1 60		-30	0	180	240	3500
P3	C2 15	C1 50		-30	0	120	180	7000
P4	C3 20	C1 45	C2 10	30	120	240	270	4000
P5	C2 15	C3 10	C1 60	30	120	240	270	3000
P6	C3 10	C2 20	C1 50	-30	0	120	180	5000

Tabla 1. Cartera de proyectos dinámica.

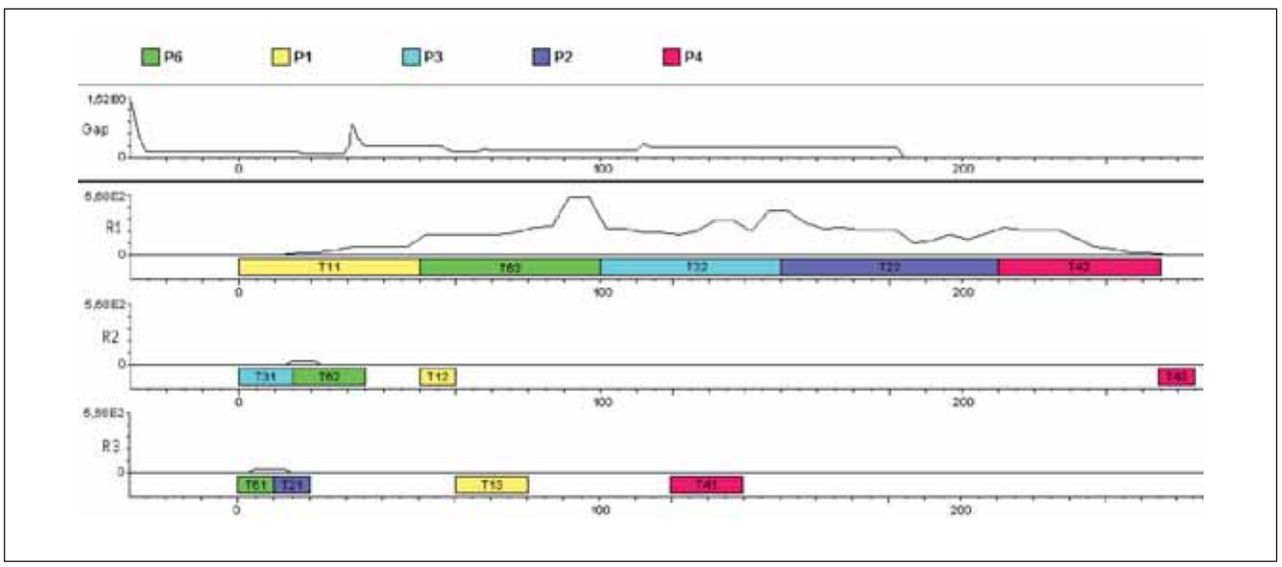

Figura 6. Evolución del sistema para el caso de estudio. Tij denota tarea j del proyecto Pi

El hecho de que en el experimento el gap de dualidad se mantiene con valores mínimos significa que las decisiones se han tomado de forma bastante eficiente (cuasi-óptima). El valor total obtenido (valor de proyectos ejecutados menos costes por retrasos y costes de recursos) es de 19402. Para ello se ha tenido que desestimar el proyecto P5. Nótese que aunque este proyecto tiene un alto valor, éste ha entrado en el sistema cuando otros proyectos, algunos de menos valor, ya estaban aceptados (P1, P2, P3 y P6) y por lo tanto no se podían rechazar. Solo se podía optar entre rechazar el P4 y el P5.

La simulación no sólo muestra la programación dinámica y el rechazo de proyectos, sino también la valoración de los recursos. Por ejemplo, en la figura 6, los precios de los slots temporales del recurso R1 son elevados en todo el periodo de planificación. Esto significa que las habilidades que posee el recurso son muy valoradas (el recurso es un cuello de botella), de esta forma si la empresa desea aumentar su ganancia realizando más proyectos de este tipo y nivelando el uso de recursos, podría intentar capacitar a los otros recursos para poder exhibir las habilidades del recurso R1, o bien introducir nuevos recursos similares al recurso R1. Por otro lado, los precios de los recursos R2 y R3 son bajos; aunque realizan algunas operaciones, están ociosos la mayor parte del tiempo.

En el caso de estudio, resulta interesante analizar el comportamiento del sistema cuando se aumenta el rango de habilidades de los recursos R2 y/o R3. En el caso de que

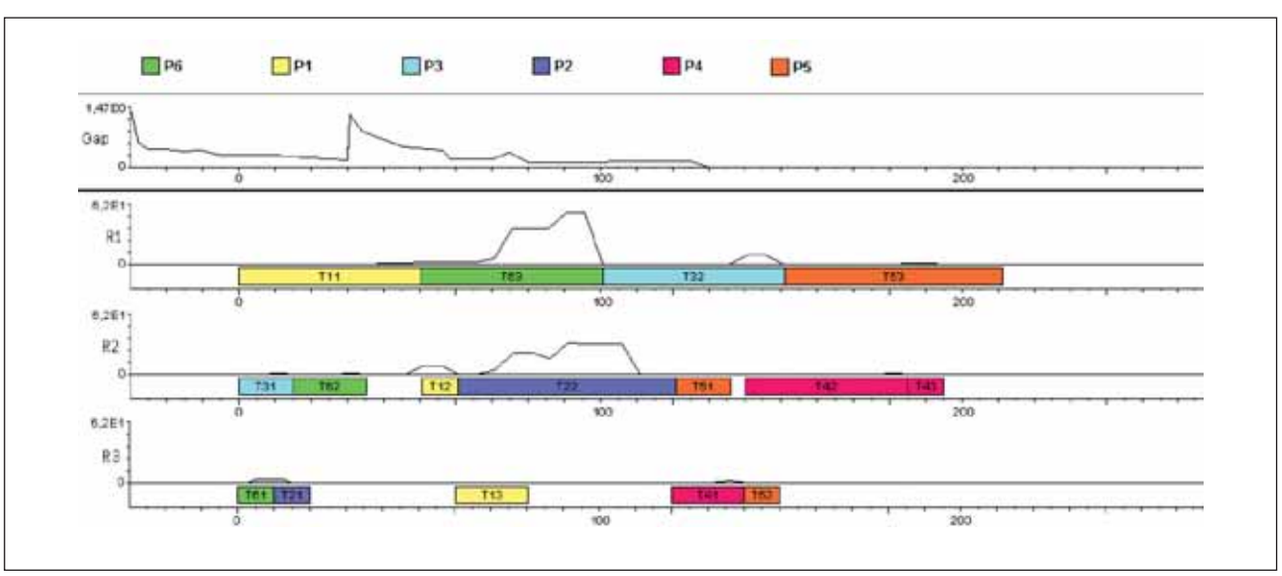

Figura 7. Evolución del sistema para el caso de estudio con habilidades del recurso R2 incrementadas.

los recursos fuesen humanos, esta capacitación adicional podría corresponderse, por ejemplo, a la asistencia de los recursos a un programa de formación. En la Figura 7 se muestra el comportamiento del sistema cuando el recurso R2 además de poseer su habilidad inicial C2, también posee la C1.

Comparando el resultado de la simulación con el del experimento previo, se puede apreciar cómo los precios de los slots temporales han aumentado en el recurso R2 y han disminuido en el recurso R1. Algunas tareas que en el experimento anterior eran ejecutadas por R1 ahora lo son por R2. Esto muestra que el sistema ha sido capaz de aprovechar la capacidad del recurso R2 para mejorar la eficiencia global. Ahora el proyecto P5 es aceptado y el beneficio total obtenido se aumenta de 19402 a 24039, un aumento superior al 23%.

6. CONCLUSIONES

Aunque la literatura sobre dirección de proyectos se ha centrado fundamentalmente en el caso de proyectos individuales, en la práctica las empresas trabajan frecuentemente con dinámicas y complejas carteras de proyectos que comparten un conjunto de recursos. A estos sistemas se los denomina entornos multiproyecto.

Para la gestión de estos sistemas, nosotros proponemos un sistema multiagente y un mecanismo programación y control on-line inspirado en mercados coordinados mediante subastas. Los proyectos necesitan completar sus tareas y por ello compiten por los recursos que exhiben las capacidades necesarias para realizar dichas tareas. A través de un proceso de subasta emergen unos precios del uso temporal de cada recurso, precios que son utilizados por los proyectos para elaborar programas propios compatibles entre si y globalmente eficientes.

En el artículo se muestra algunas de las posibilidades que la aproximación propuesta ofrece para el soporte a la toma de decisión en entornos multiproyecto. El sistema asigna recursos a las tareas de los proyectos dinámicamente, y ayuda además a la toma de decisiones de si se debe rechazar algún proyecto teniendo en cuenta para ello el valor aportado por los proyectos y el estado del sistema.

También se muestra cómo los precios aportan información sobre la criticidad que tienen los diferentes recursos para el desempeño de la cartera de proyectos. Los precios permiten valorar en tiempo real, si se debe adquirir más recursos de un tipo durante un cierto periodo de tiempo, o si se debe tratar de dotar con ciertas capacidades adicionales a ciertos recursos.

Nuestra aproximación contribuye a rellenar el hueco de literatura existente entre la gestión de cartera de proyectos –generalmente centrada en estrategia corporativa y finanzas– y el trabajo en dirección de proyectos –fundamentalmente dedicado a aspectos operacionales como la asignación de recursos y la programación–.

El objeto último de nuestro trabajo es conseguir un sistema operativo en entornos reales genéricos. Para ello sería necesario relajar algunos de los supuestos simplificadores incluidos para realizar el presente prototipo. Se debería incorporar entre otras mejoras, la posibilidad de establecer hitos intermedios en la ejecución de los proyectos e introducir la subcontratación como una alternativa al uso de recursos propios. Además sería interesante modificar el procedimiento de subasta, para comprobar el impacto que tiene ciertos aspectos de este procedimiento en el funcionamiento global del sistema. Todo ello constituye líneas de investigación en las que actualmente estamos trabajando.

7.- AGRADECIMIENTOS

Agradecimientos: Los Ministerios de Educación y Ciencia y de Ciencia e Innovación han financiado la realización de este trabajo dentro de los proyectos referencia Ref.: DPI2005-05676 y Ref: TIN2008-06464-C03-02.

8.- BIBLIOGRAFÍA:

- Anavi-Isakow S, Golany B. "Managing multi-project environments through constant work-in-process". International Journal of Project Management. 2003, Vol.21-1, p.9-18.
- Araúzo JA, Galán JM, López-Paredes A, et al. "Multi-agent technology for scheduling and control projects in multi-project environments. An Auction based approach". IBERAGENTS 08.7th Ibero-American Workshop in Multi-Agent Systems. Lisboa: 2008.
- Cabac L. "Multi-agent system: A guiding metaphor for the organization of software development projects". En: Carbonell JG, Siekman J (eds). Multiagent System Technologies. Lecture Notes in Artificial Intelligence Vol. 4687. Berlin / Heidelberg: Springer, 2007. p.1-12.
- Clearwater SH. Market-Based Control: A Paradigm for Distributed Resource Allocation. River Edge: World Scientific, 1996. 311p. ISBN: 981-022254-8
- Cohen I, Mandelbaum A, Shtub A. "Multi-Project Scheduling and Control: A Process-Based Comparative Study of the Critical Chain Methodology and Some Alternatives". Project Management Journal. 2004, Vol.35, p.39-49.

- Confessore G, Giordani S, Rismondo S. "A market-based multi-agent system model for decentralized multi-project scheduling". Annals of Operations Research. 2007, Vol.150-1, p.115-135.
- Galán JM, del Olmo R, López-Paredes A. "Diffusion of Domestic Water Conservation Technologies in an ABM-GIS Integrated Model ". En: Corchado E, Abraham A, Pedrycz W (eds). Hybrid Artificial Intelligence Systems. Lecture Notes in Artificial Intelligence Vol. 5271. p.567-574. Springer-Verlag, 2008.
- Goldratt EM. Critical Chain. Great Barrington: North River Press, 1997. 246p. ISBN: 9780884271536
- Hans EW, Herroelen W, Leus R, et al. "A hierarchical approach to multi-project planning under uncertainty". Omega. 2007, Vol.35-5, p.563-577.
- Jennings NR, Wooldridge MJ. "Applying agent technology". Applied Artificial Intelligence. 1995, Vol.9-4, p.357-369.
- Kao HP, Wang B, Dong J, et al. "An event-driven approach with makespan/cost tradeoff analysis for project portfolio scheduling". Computers in Industry. 2006, Vol.57-5, p.379-397.
- Kim K, Paulson J. "Multi-agent distributed coordination of project schedule changes". Computer-Aided Civil and Infrastructure Engineering. 2003a, Vol.18-6, p.412-425.
- Kim K, Paulson J. "Agent-based compensatory negotiation methodology to facilitate distributed coordination of project schedule changes". Journal of Computing in Civil Engineering. 2003b, Vol.17-1, p.10-18.
- Kim K, Paulson J, Levitt RE, et al. "Distributed coordination of project schedule changes using agent-based compensatory negotiation methodology". Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM. 2003, Vol.17-2, p.115-131.
- Kim K, Paulson J, Petrie J. "Agent-based project schedule coordination". Computing in Civil and Building Engineering. 2000, Vol.1, p.732-739.
- Kumara SRT, Lee YH, Chatterjee K. "Distributed multiproject resource control: A market-based approach". CIRP Annals - Manufacturing Technology. 2002, Vol.51-1, p.367-370.
- Lee YH, Kumara SRT, Chatterjee K. "Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism". Journal of Intelligent Manufacturing. 2003, Vol.14-5, p.471-484.
- Lopez-Paredes A, Sauri D, Galan JM. "Urban Water Management with Artificial Societies of Agents: The FIRMABAR Simulator". SIMULATION. 2005, Vol.81-3, p.189-199.
- Luh PB, Hoitomt DJ. "Scheduling of manufacturing systems using the Lagrangian relaxation technique". IEEE Transactions on Automatic Control. 1993, Vol.38-7, p.1066-1079.
- Martínez J, Pérez M, Martínez A. "La organización de las oficinas técnicas de proyectos de ingeniería". DYNA Ingeniería e Industria. 2001, Vol.76-1, p.7-11.
- Pajares J, Hernandez-Iglesias C, López-Paredes A. "Modelling Learning and R&D in Innovative Environments: a Cognitive Multi-Agent Approach". Journal of Artificial Societies and Social Simulation. 2004, Vol.7-2. http://jasss.soc.surrey.ac.uk/7/2/7.html.
- Pascual JA, Pajares J, Lopez-Paredes A. "Explaining the Statistical Features of the Spanish Stock Market from the Bottom-Up". En: Bruun C (ed). Advances in Artificial Economics. The Economy as a Complex Dynamic System. Lecture Notes in Economics and Mathematical Systems Vol. 584. Springer, 2006.
- Pavón J, Pérez JL. Agentes Software y Sistemas Multi-Agente. Madrid: Pearson Educacion, 2005. 286p. ISBN: 9788420543673.
- Posada M, López-Paredes A. "How to Choose the Bidding Strategy in Continuous Double Auctions: Imitation Versus Take-The-Best Heuristics". Journal of Artificial Societies and Social Simulation. 2008, Vol.11-1, p.6. http://jasss.soc.surrey.ac.uk/11/1/6.html.
- Wang J, Luh PB, Zhao X, et al. "An optimization-based algorithm for job shop scheduling". Sadhana - Academy Proceedings in Engineering Sciences. 1997, Vol.22-2, p.241-256.
- Wooldridge MJ. An Introduction to Multiagent Systems. New York: John Wiley & Sons, 2002. 348p. ISBN: 9780471496915.
- Wu S, Kotak D. Agent-based collaborative project management system for distributed manufacturing. IEEE International Conference on Systems, Man and Cybernetics, 2003. Vol.2, p.1223-1228.
- Yan Y, Kuphal T, Bode J. "Application of multiagent systems in project management". International Journal of Production Economics. 2000, Vol.68-2, p.185-197.
- Zhao X, Luh PB, Wang J. "Surrogate Gradient Algorithm for Lagrangian Relaxation". Journal of Optimization Theory and Applications. 1999, Vol.100-3, p.699-712.