# Middleware cataláctico para mercados Grid descentralizados

## Catallactic Grid-enabled Middleware Based on a Free-market Approach

O. Ardaiz, P. Chacín, I. Chao et al.

Resumen

En este artículo describimos un prototipo que utiliza *middleware* cataláctico. Este *middleware* está basado en el concepto de *Catallaxy* entendido como un mercado libre que se autoorganiza tal como lo describió von Hayek [7], que entendía el mercado como un mecanismo descentralizado de coordinación opuesto a una economía planificada de manera centralizada. La implementación hace uso del *toolkit de Globus,* JXTA y WSRF.

**Palabras claves:** Middleware, mercados grid, asignación económica

Summary

In this paper we describe an application deployment using a Catallactic Grid-enabled middleware, which is based on the Catallaxy "free market" self-organisation approach described by von Hayek [7], who understood the market as a decentralised coordination mechanism opposite to a centralised command economy. The implementation makes use of Globus Toolkit, JXTA and WSRF.

**Keywords:** Middleware, Grid Markets, Economic-based Allocation

El concepto de *Catallaxy* se entiende como un mercado libre que se autoorganiza

## 1. Introducción

Ha habido un interés significativo en utilizar paradigmas económicos para intercambiar recursos y servicios en el Grid [3]. La principal motivación es la capacidad para planificar el acceso a los servicios basados en un mecanismo de mercado (tales como subastas), permitiendo un acceso más justo y eficiente a recursos compartidos de gran demanda. Otras propuestas existentes utilizan un *broker* centralizado que coordina el acceso a los recursos. Nuestra propuesta se basa en el mecanismo de Catallaxy propuesto por von Hayek [7], que no necesita ninguna institución central para organizar el mercado. La Catallaxy se basa en la autoorganización del "mercado libre", que utiliza solamente el ajuste de precios dentro del mercado para asignar demandas con servicios.

En un Grid aparecen dos mercados interrelacionados: el de recursos y el de servicios

La *Catallaxy* es un mecanismo de coordinación para sistemas que consiste en agentes descentralizados autónomos y se basa en continuas negociaciones entre ellos, de forma que los precios que son intercambiados funcionan como señales que se distribuyen por el sistema logrando la coordinación del mismo [4]. *Catallaxy* es una manera de informar al individuo (agente) sobre el conocimiento que pueden contener otros agentes al mismo tiempo que se proporciona un intercambio de la información que conduce a la generación de precios los cuales se conforman con el valor que cada agente asigna a la información respectiva [2]. *Catallaxy* por lo tanto conduce al desarrollo de individuos que se auto-organizan, de tal modo que los sistemas se comportan en una manera *Peer-2-Peer*.

### 1.1. *Catallaxy* y mercados Grid

Los mecanismos del "libre-mercado" de la *Catallaxy* se pueden aplicar al Grid. En un Grid aparecen dos mercados interrelacionados: el de recursos y el de servicios. En el mercado de recursos, los proveedores venden su capacidad de cómputo, almacenamiento, ancho de banda o herramientas a los compradores de recursos. El producto negociado son los recursos físicos que serán utilizados por los compradores para ejecutar sus programas. En el mercado de recursos hay una gran cantidad de vendedores y de compradores, por lo tanto se puede establecer un mercado libre donde se puede aplicar la *Catallaxy*.

En el mercado de servicios los proveedores venden servicios a clientes. El producto negociado son los servicios que proporcionan una particular funcionalidad: servicios de transcodificación, de búsqueda en bases de datos, de análisis de moléculas, etc. El comprador del mercado de servicios está interesado en usar una aplicación particular, no en ejecutar su propio código de aplicación. En el mercado de servicios hay una gran cantidad de vendedores y de compradores, por lo tanto también se puede establecer un mercado libre donde se puede aplicar la *Catallaxy*.

Por otra parte, los proveedores de servicios pueden comprar recursos en el mercado de recursos Grid para proporcionar servicios en el mercado de servicios Grid; por ejemplo, un proveedor de servicio de transcodificación puede comprar recursos de cómputo en el mercado de recursos para ejecutar su programa de transcodificación para una petición particular del cliente. Ambos mercados son dependientes el uno del otro, pero funcionan de manera autónoma usando los mecanismos catalácticos como ha sido demostrado por simulaciones [2].

### 1.2. Aplicación de demostración

Cat-COVITE se basa en una aplicación que sirve para integrar actividades de diseñadores de producto, ingenieros de especificación y usuarios en la industria de ingeniería y construcción

El prototipo Cat-COVITE se basa en una aplicación anterior que sirve para integrar actividades de diseñadores de producto, ingenieros de especificación y usuarios en la industria de ingeniería y construcción [10]. La aplicación existente implica la búsqueda en bases de datos distribuidas de los productores de componentes con los que diseñar el nuevo producto. El modelo adoptado en la aplicación Cat-COVITE es replicable en un número significativo de otras aplicaciones industriales que hagan uso de bases de datos distribuidas.

### 1.3. Introducción al WS-Agreement

La especificación del protocolo WS-AGreement ha sido desarrollada por el grupo de trabajo en el protocolo de planificación y asignación de recursos (GRAAP) del área SRM (gestión de recursos y planificación) del GGF (Global Grid Forum) [8]. El WS-Agreement es un protocolo que utiliza XML como lenguaje para especificar un acuerdo entre un proveedor y un consumidor [13] de recursos o servicios. Este protocolo sólo involucra la aceptación o rechazo del acuerdo y no se piensa directamente en apoyar la negociación, aunque puede formar una base para desarrollarla. El WS-Agreement se utiliza en Cat-COVITE para elegir entre múltiples proveedores de servicios y recursos. El proveedor de servicio actúa como WS-Agreement Provider, mientras que el consumidor del servicio actúa como WS-Agreement Initiator. La sección 3.2 detalla los conceptos y los requisitos del WS-Agreement en el contexto del prototipo de Cat-COVITE.

## 2. *Middleware* cataláctico

La implementación de la Catallaxy en Grids requiere el diseño de un *middleware* cataláctico que ofrezca un sistema de mecanismos genéricos de negociación

La implementación de la *Catallaxy* en Grids requiere el diseño de un *middleware* cataláctico que ofrezca un sistema de mecanismos genéricos de negociación, permitiendo estrategias especializadas y políticas que se agregarán dinámicamente como *plugins*. Se pretende que el *middleware* ofrezca un conjunto de abstracciones y mecanismos de alto nivel para localizar y gestionar recursos, para localizar otros agentes que negocian y establecer contratos entre agentes, adaptándose a condiciones cambiantes. Los problemas técnicos que necesitan ser evaluados se tratan en esta sección.

### 2.1. Requisitos

- **Escalabilidad en entornos altamente dinámicos.** El *middleware* cataláctico debe poder utilizarse en entornos con millares de nodos, en un ambiente altamente dinámico donde los nodos se

incorporan y salen de la red con frecuencia. El dinamismo implica que la información sobre el sistema se debe mantener en un mínimo y que las actualizaciones deben ser fáciles y eficientes.
- **Soportar entornos heterogéneos.** La escala también implica un alto nivel de heterogeneidad en los usos, la plataforma subyacente, recursos, características del servicio de proveedores, y disponibilidad de nodos.
- **Compatibilidad con diversas plataformas base.** Para ello se debería de definir un API genérico.
- **Auto-organización.** El dinamismo de la red previene de una configuración a priori de los nodos o el mantenimiento centralizados de los archivos de configuración. Los nodos necesitan descubrir continuamente las características de la red y adaptarse por consiguiente, requiriendo la distribución de algunas funciones importantes del sistema como seguridad, gestión de recursos, topología, entre otros, que se han reservado tradicionalmente a nodos muy especializados.

## 2.2. Conceptos y pautas de diseño

Se ha adoptado un paradigma P2P que facilita las siguientes características:

- Descentralización, no hay un solo punto centralizado de coordinación o administración de los nodos.
- Las interacciones entre nodos son simétricas, todos los nodos son simultáneamente clientes y servidores que solicitan y que proporcionan servicios.
- La topología es no determinista, en cualquier momento, la topología de la red P2P es totalmente impredecible.
- El conjunto de nodos que conforma la red puede variar constantemente.
- La asignación dinámica y virtual de las trayectorias de comunicación se crean dinámicamente basadas en varios factores, tales como la conjunción de la red o el estado de los nodos intermedios.

Una arquitectura en capas proporciona una separación de funcionalidad

## 2.3. Arquitectura

Una arquitectura en capas proporciona una separación de funcionalidad. Facilita la construcción de un sistema más adaptable, las capas superiores se pueden especializar progresivamente (mediante reglas y estrategias enchufables) en dominios específicos del uso. Se han diseñado las siguientes capas:

1. **Capa de aplicación.** Dada por el dominio específico de aplicación como herramientas de colaboración, entorno de resolución de problemas y muchos otros. Las aplicaciones usan la plataforma inferior para funciones como la comunicación y la gestión de recursos. Sin embargo, algunas aplicaciones pueden tener recursos a nivel de aplicación, como un cuarto de reunión virtual en una herramienta de colaboración o un algoritmo determinado en un entorno científico. El modelo de interacción entre la capa de aplicación y el *middleware* es dependiente de la aplicación y del *middleware*.
2. **Capa de algoritmos económicos.** Implementa los algoritmos económicos para la asignación de recursos. Estos algoritmos deben ser independientes del dominio y la plataforma. Esta capa incluye un conjunto de agentes de servicios que desempeñan los papeles de vendedores y de compradores en los mercados de servicios y de recursos. También en esta capa están las extensiones y las especializaciones de las funcionalidades proporcionadas por el marco inferior, para adaptarlas al dominio específico y a las políticas de asignación de recurso del lugar.
3. **Capa del marco económico.** Ofrece las primitivas que permiten la realización de los algoritmos catalácticos, tales como encontrar agentes para negociar, comenzar la negociación, hacer una oferta, etc. Es dependiente de la plataforma de agente que se utilice, pero debe de ser independiente del dominio del aplicación y de la plataforma base. Esta capa se estructura como un conjunto de entidades básicas que modelan la interacción de los agentes que negocian en un mercado para intercambiar mercancías. Estas entidades abstractas son los bloques con los que se construyen los algoritmos catalácticos.

La capa de aplicación está dada por el dominio específico de aplicación como herramientas de colaboración, entorno de resolución de problemas y muchos otros

4. **Capa de agentes P2P.** Plataforma que soporta los agentes catalácticos. Ofrece un modelo genérico de aplicación P2P con abstracciones para el descubrimiento y la comunicación, y una interfaz genérica con la plataforma inferior. Esta capa cubre las funciones básicas que serán utilizadas por todas las implementaciones.
5. **Capa de plataforma base.** Soporta las aplicaciones y el *middleware* cataláctico. Es (potencialmente) específica del dominio. El modelo de interacción con el *middleware* cataláctico depende de la arquitectura de la plataforma base, pero en general requerirá la implementación de un conector, que encamine la petición de recursos a los agentes económicos correspondientes. En algunos casos, esto puede incluso requerir la re-implementación de algunos componentes de la plataforma base, como los GRAM (gestores de asignación de recursos) en Globus [6].

## 3. Aplicación CAT-COVITE

La aplicación COVITE se divide en dos servicios funcionales: de seguridad y de búsqueda en múltiples bases de datos

En el proyecto original [10], proveedores y contratistas colaboran para obtener productos para un proyecto de construcción particular usando la aplicación COVITE. Estos proyectos son generalmente únicos, muy complejos e implican a muchos participantes de un gran número de organizaciones que colaboran. Estos participantes trabajan concurrentemente, requiriendo la colaboración en tiempo real entre participantes geográficamente alejados. Cada consorcio es en efecto una organización virtual (VO). La aplicación permite hacer búsquedas en una gran cantidad de bases de datos de proveedores para recuperar los productos que cumplen los criterios fijados por los contratistas. La aplicación permite realizar búsquedas, haciendo uso de un cluster de máquinas en un Grid.

La aplicación COVITE se divide en dos servicios funcionales: servicio de seguridad y servicio de búsqueda en múltiples bases de datos (MDSS). El MDSS soportado por el Grid permite buscar en una gran cantidad de bases de datos de proveedores (SD), usando una instancia de un servicio principal de Grid (MGS). En este caso, la petición se define según un modelo de datos que son específicos de un dominio dado de aplicación; las peticiones arbitrarias (como en el motor de búsqueda de Google.com, por ejemplo) no se permiten. La aplicación COVITE permite a las VO planificar, programar, coordinar, y compartir componentes entre los diseñadores y diversos proveedores.

### 3.1. Prototipo Cat-COVITE y los mercados Grid

El servicio complejo de MGS es la entidad compradora en el mercado de servicios

Los diferente componentes de la aplicación COVITE se pueden mapear a los agentes de los mercados Grid catalácticos. La aplicación Cat-COVITE resultante permitirá al cliente acceder a mayor cantidad de servicios y recursos de una manera más eficiente. La figura 1 muestra los componentes de Cat-COVITE y los agentes catalácticos compradores y vendedores en los mercados Grid de servicios y de recursos. Cat-COVITE se compone de tres elementos principales, el servicio principal de Grid, MGS (un tipo de servicio complejo), el servicio de ejecución de peticiones de datos, Query Job Service (un tipo de servicio básico), y los recursos de ejecución de peticiones (recursos de cómputo).

El servicio complejo de MGS es la entidad compradora en el mercado de servicios; y el servicio básico de ejecución de peticiones de datos, Query Job Service, es la entidad vendedora en el mercado de servicios. Un servicio complejo de MGS incluye las actividades siguientes:

- Traducir una petición a un servicio básico.
- Negociar paralelamente con un número de agentes que representan los servicios de ejecución de petición de búsqueda datos (servicios básicos).
- Enviar peticiones a una lista de servicios de ejecución de búsqueda de datos (servicios básicos).

El servicio básico de la ejecución de búsquedas de datos implica la ejecución de la petición en una base de datos particular y consiste en:

- La ejecución de la petición de búsqueda de datos.
- La traducción de la petición a los requisitos de recurso.

Dentro de la aplicación de Cat-COVITE, el servicio básico de la ejecución de peticiones de búsqueda de datos tiene que proporcionar el tiempo de respuesta y la calidad necesaria. Con este objetivo, el servicio básico de la ejecución de peticiones de búsqueda de datos compra recursos en el mercado del recurso. Las entidades proveedoras de recursos pueden proporcionar recursos vía el gestor de recursos locales (LRM). Los agentes del recurso actúan en nombre de estos LRM, que ocultan los recursos físicos detrás de ellos.

El servicio básico de ejecución de peticiones de búsqueda de datos es la entidad compradora en el mercado de recursos y los gestores locales LRM la entidad vendedora. Las funcionalidades principales del agente del servicio básico en el mercado del recurso son:

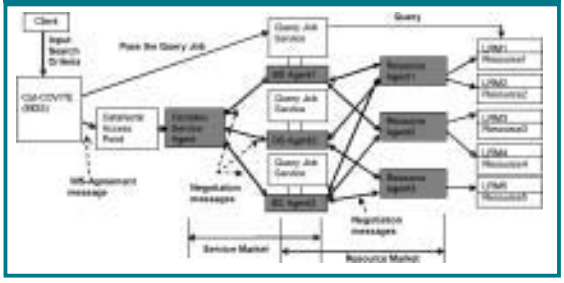
- La coasignación de los recursos (agregaciones de recursos) a través de negociaciones paralelas con diversos proveedores (agentes de los gestores locales de recursos).
- La información al servicio básico sobre el resultado de la negociación del recurso.

El servicio básico de la ejecución de peticiones de búsqueda de datos de Cat-COVITE tiene que proporcionar el tiempo de respuesta y la calidad necesaria

## 3.2. Uso de WS-Agreement en Cat-COVITE

Un acuerdo consiste en varias partes, según el borrador del protocolo WS-Agreement [13]: la sección del nombre del acuerdo, que es opcional; el contexto del acuerdo que incluye las partes de un acuerdo, las referencias a servicios proporcionados para soportar el acuerdo, y su duración; los términos del acuerdo, que describen el acuerdo en sí mismo; pueden contener a su vez: los términos de la descripción del servicio, la información que proporcionan y la identificación de un servicio al cual este acuerdo pertenece. Y los términos del servicio, que especifican los porcentajes de disponibilidad y calidad del servicio que están conviniendo.

**FIG. 1: MERCADOS GRID EN CAT-COVITE Y AGENTES CATALÁCTICOS**



El ejemplo de uso propuesto en Cat-COVITE para la interacción entre la aplicación y el *middleware* es: "El MGS necesita ejecutar una petición de búsqueda de datos, envía al CAP una oferta de acuerdo (AO) basada en la plantilla de acuerdos (EN) descargada del punto de acceso cataláctico (CAP) para encontrar un servicio de ejecución de peticiones de búsqueda. El agente de servicio complejo, actuando en nombre del servicio complejo (MGS) elegido por el CAP, negocia con los agentes de servicio básico por servicios de ejecución de peticiones de búsqueda". La plantilla de acuerdo (EN) especifica los elementos de la descripción del servicio que son permitidos por el punto de acceso. Hemos creado una plantilla específica de acuerdo EN para la aplicación de Cat-COVITE. La oferta de acuerdo (AO) la inicia el iniciador de acuerdo (MGS). La aceptación del acuerdo es igual que su oferta, si el proveedor del acuerdo acepta sus condiciones. Si el proveedor del acuerdo no acepta la oferta, el iniciador tiene que enviar otra. Un ejemplo de oferta de acuerdo creado a partir de la plantilla especifica de Cat-COVITE es:

La plantilla de acuerdo (EN) especifica los elementos de la descripción del servicio que son permitidos por el punto de acceso

```
<?xml version="1.0" encoding="UTF-8"?>
<AgreementOfferLite>
  <Name>QueryServiceTemplateLite</Name>
  <Context>
    <AgreementInitiator>
      <Name>Your Name</Name>
    </AgreementInitiator>
  </Context>
  <Terms>
    <Executable> SELECT IDProduct, ManufacturerName,
.....Price FROM Product ORDER BY Price DESC
    </Executable>
    <PayForService>100</PayForService>
  </Terms>
</AgreementOfferLite>
```

## 4. Implementación del prototipo

### 4.1. Implementación del *middleware*

El *middleware* está diseñado como un sistema de agentes simples especializados, que se implementan con agentes ligeros de la plataforma DIETS [5]. Los agentes de la capa marco proveen las funciones básicas que se necesitan para implementar los algoritmos económicos, como el acceso a los mercados, a las negociaciones, a las mercancías negociadas, a los agentes que negocian, etc. Los agentes de la capa de agentes *Peer-to-Peer* ofrecen las funcionalidades de bajo nivel para soportar el marco: formación de red *overlay,* descubrimiento de objetos y comunicaciones. Una descripción detallada de la implementación del middleware se encuentra en [1].

Las funciones de formación de red *overlay*, descubrimiento de objetos y comunicaciones se implementan usando el protocolo Peer Resolver de JXTA [12]. La gestión de recursos locales, en este caso los ofrecidos por los proveedores de servicio, se basa en el marco de WSRF [14] ofrecido por Globos Toolkit v4.

El *middleware* está diseñado como un sistema de agentes simples especializados, que se implementan con agentes ligeros de la plataforma DIETS

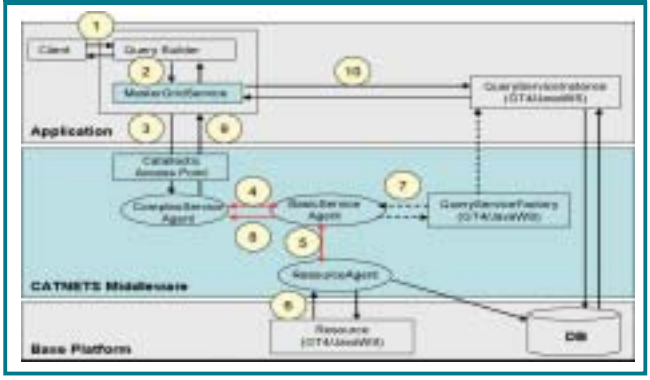
Las búsquedas de servicios Grid son propagadas por el mecanismo descentralizado implementado por el protocolo Peer Resolver de JXTA, permitiendo realizar peticiones multiatributo complejas. Los nodos remotos registrados previamente como revolvedores de un tipo de petición utilizarán el servicio de índices de GT4 para resolver la petición de búsqueda de determinados atributos. En una primera implementación este mecanismo permite la búsqueda del servicio básico y necesitará ser extendido y modificado para obtener el desempeño adecuado, especialmente referente a los algoritmos de búsqueda sobre la red de overlay de JXTA.

### 4.2. Integrando Cat-COVITE con el *middleware* cataláctico

Aquí describimos cómo la aplicación y el *middleware* pueden ser integrados. La figura 2 representa la arquitectura de alto nivel, identificando la colocación de componentes lógicos en tres capas: la capa de aplicación, la de middleware cataláctico y la plataforma base. En la capa de aplicación, esta tiene que conectar con la interfaz del *middleware* que debe permitir efectuar peticiones de servicios al *middleware* y utilizar las referencias a las instancias de servicio proporcionados por el *middleware* para ejecutar tales servicios. En la capa del *middleware*, un sistema de agentes proporciona las capacidades para negociar por los servicios y los recursos necesarios para ejecutarlos. El agente complejo de servicio que actúa en nombre de la aplicación inicia la negociación. Los agentes del servicio básico y del recurso realizan la negociación por los servicios y los recursos, respectivamente. También, una factoría de servicios proporciona la instanciación de servicios en el entorno de ejecución seleccionado durante el proceso de la negociación. Finalmente, en la capa de la plataforma base, un recurso se crea para gestionar la asignación de recursos al servicio. Este recurso representa el "estado" del servicio desde la perspectiva del *middleware*.

En la capa del *middleware*, un sistema de agentes proporciona las capacidades para negociar por los servicios y los recursos necesarios para ejecutarlos

**FIG. 2: INTEGRACIÓN DEL MIDDLEWARE CATALÁCTICO CON LA APLICACIÓN CAT-COVITE**



El flujo de información entre los componentes lógicos se puede resumir como sigue: un cliente realiza una petición a la aplicación (1), que construye una pregunta y solicita la ejecución de la pregunta al servicio principal de Grid, MGS (2). El MGS entra en contacto con un punto de acceso cataláctico que pide una plantilla de WS-Agreement para tal servicio. El MGS completa la plantilla y envía una oferta de acuerdo (3). El agente complejo del servicio inicia los mecanismos catalácticos para encontrar servicios básicos y recursos apropiados. El agente de servicios complejos utiliza los mecanismos de descubrimiento implementados en la capa de agentes *Peer-to-Peer* para localizar agentes de servicios básicos que provean servicios de ejecución de peticiones de búsqueda. Cuando se descubre un número de agentes de servicio básico, comienzan las negociaciones con uno de ellos (4). Por su parte el agente de servicio básico debe descubrir y negociar con un agente de recursos para obtener los recursos para la ejecución de la petición en el mercado (5). Las negociaciones son implementadas por la capa económica del marco, donde se pueden utilizar diversos protocolos dependiendo de la estrategia del agente. Cuando se alcanza un acuerdo con un agente de servicio básico, el del recurso solicita un recurso para monitorizar los asignados y devuelve al agente de servicio básico una referencia a este recurso (6). A continuación los agentes del servicio básico utilizan la factoría del servicio de peticiones de búsqueda para solicitar el servicio de búsqueda en el contendor GT4 seleccionado (7). Los agentes del servicio básico devuelven al agente de servicio complejo la referencia al servicio recientemente solicitado (8). La referencia al servicio de peticiones de búsqueda se devuelve al MSG (9), que lo utiliza para invocar el servicio, pasando la pregunta que se ejecutará (10).

La actual especificación del protocolo WS-Agreement no es suficiente para utilizarse con protocolos de negociación complejos

## 5. Conclusiones

Se ha presentado un *middleware* cataláctico basado en los conceptos de Catallaxy o mercado libre. También se ha descrito la implementación de una aplicación prototipo que utiliza ese *middleware*. La implementación se basa en estándares y código libre como Globus Toolkit, JXTA y WSRF.

Como conclusiones de la implementación y pruebas realizadas, podemos destacar que la actual especificación del protocolo WS-Agreement no es suficiente para utilizarse con protocolos de negociación complejos. También la especificación de WSRF es todavía muy general y no ofrece una manera sencilla de gestionar recursos virtuales.

Los próximos pasos de nuestro trabajo incluyen el desarrollo de componentes para realizar medidas de desempeño, y la utilización del *middleware* como soporte a una aplicación diferente con el objetivo de validar el *middleware*.

La especificación de WSRF es todavía muy general y no ofrece una manera sencilla de gestionar recursos virtuales

### Reconocimientos

Este trabajo fue apoyado en parte por la Unión Europea bajo el contrato CATNETS IST-FP6-003769, y el Gobierno Español bajo contrato TIC2002-04258-C03-01.

**Óscar Ardaiz**
(oscar.ardaiz@unavarra.es)
Dpto. de Matemáticas e Informática
Universidad Pública de Navarra
**Pablo Chacín, Isaac Chao**
(pchacin@ac.upc.edu), (ichao@ac.upc.edu),
**Félix Freitag, Leandro Navarro**
(felix@ac.upc.edu), (leandro@ac.upc.edu)
Dept. Arquitectura de Computadores - UPC

## Referencias


[1] Ardaiz O., Chacín P., Chao I., Freitag F., Navarro L. *An Architecture for Incorporating Decentralized Economic Models in Application Layer Networks*. Smart Grids Technologies Workshop, Utrecht, Holland, 2005.

[2] Ardaiz O., Artigas P., Eymann T., Freitag F., Navarro L., Reinicke M. *The Catallaxy Approach for Decentralized Economic-based Allocation in Grid Resource and Service Markets*. Special Issue on Agent-based Grid Computing, International Journal of Applied Intelligence, accepted.

[3] Buyya R., Abramson D., Giddy J., and Stockinger H. *Economic Models for Resource Management and Scheduling in Grid Computing*. The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, May 2002

[4] CATNET Project deliverables. D3: Catallaxy Evaluation Report. March 2003. Available at: *http://research.ac.upc.es/catnet/pubs/D3.pdf*

[5] Diet Agents, *http://diet-agents.sourceforge.net/*, Oct. 2005.

[6] Foster I., Kesselman C., Lee C., Lindell R., Nahrstedt K., Roy A. *A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation*. Intl Workshop on Quality of Service, 1999.

[7] Hayek F. A., Bartley W., Klein P., Caldwell B., *The Collected Works of F. A. Hayek*. University of Chicago Press, 1989.

[8] Global Grid Forum (2005), *www.ggf.org/*

[9] Globus Toolkit version 4.0, *www.globus.org*

[10] Joita L., Pahwa J. S., Burnap P., Gray A., Rana O., and Miles J. *Supporting Collaborative Virtual Organisations in the Construction Industry Via the Grid*. Proceedings of the UK e-Science All Hands Meeting 2004, 31st Aug.-3rd Sept. 2004 Nottingham, UK.

[11] Joseph J:, Ernest M., Fellenstein C. *Evolution of Grid Computing Architecture and Grid Adoption Models*. IBM Systems Journal, Volume 43, Issue 4 (January 2004)

[12] Traversat, Abdelaziz, and Pouyoul, Project JXTA: *Loosely-Consistent DHT Rendezvous Walker*. Sun Microsystems, Inc., *www.jxta.org/project/www/docs/jxtadht.*

[13] Web Services Agreement Specification (WS-Agreement), 28 June 2005 *https://forge.gridforum.org/docman2/ViewCategory.php?group_id=71&category_id=659*

[14] WS-Resource Framework, *www.globus.org/wsrf/*