

*Revista*  
*Española de*  
**Innovación,**  
**Calidad e**  
**Ingeniería del Software**



Volumen 7, No. 3, diciembre, 2011

Web de la editorial: [www.ati.es](http://www.ati.es)

Web de la revista: [www.ati.es/reicis](http://www.ati.es/reicis)

E-mail: [calidadsoft@ati.es](mailto:calidadsoft@ati.es)

ISSN: 1885-4486

Copyright © ATI, 2011

Ninguna parte de esta publicación puede ser reproducida, almacenada, o transmitida por ningún medio (incluyendo medios electrónicos, mecánicos, fotocopias, grabaciones o cualquier otra) para su uso o difusión públicos sin permiso previo escrito de la editorial. Uso privado autorizado sin restricciones.

Publicado por la Asociación de Técnicos de Informática (ATI), Via Laietana, 46, 08003 Barcelona.

Secretaría de dirección: ATI Madrid, C/Padilla 66, 3º dcha., 28006 Madrid



# **Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS)**

## **Editor**

**Dr. D. Luís Fernández Sanz (director)**

Departamento de Ciencias de la Computación, Universidad de Alcalá

## **Miembros del Consejo Científico**

**Dr. Dña. Idoia Alarcón**

Depto. de Informática  
Universidad Autónoma de Madrid

**Dr. D. José Antonio Calvo-Manzano**

Depto. de Leng y Sist. Inf. e Ing. Software  
Universidad Politécnica de Madrid

**Dra. Tanja Vos**

Depto. de Sist. Informáticos y Computación  
Universidad Politécnica de Valencia

**Dña. M<sup>a</sup> del Pilar Romay**

CEU Madrid

**Dr. D. Alvaro Rocha**

Universidade Fernando Pessoa  
Porto

**Dr. D. Oscar Pastor**

Depto. de Sist. Informáticos y Computación  
Universidad Politécnica de Valencia

**Dra. Dña. María Moreno**

Depto. de Informática  
Universidad de Salamanca

**Dra. D. Javier Aroba**

Depto de Ing. El. de Sist. Inf. y Automática  
Universidad de Huelva

**D. Guillermo Montoya**

DEISER S.L.  
Madrid

**Dr. D. Pablo Javier Tuya**

Depto. de Informática  
Universidad de Oviedo

**Dra. Dña. Antonia Mas**

Depto. de Informática  
Universitat de les Illes Balears

**D. Jacques Lecomte**

Meta 4, S.A.  
Francia

**Dra. Raquel Lacuesta**

Depto. de Informática e Ing. de Sistemas  
Universidad de Zaragoza

**Dra. María José Escalona**

Depto. de Lenguajes y Sist. Informáticos  
Universidad de Sevilla

**Dr. Dña. Aylin Febles**

CALISOFT  
Universidad de Ciencias Informáticas (Cuba)

---

## Contenidos

---

**REICIS**

|   |           |
|---|-----------|
| <b>Editorial</b>  | <b>4</b>  |
| <i>Luis Fernández-Sanz</i>  |           |
| <b>Presentación</b>   | <b>5</b>  |
| <i>Luis Fernández-Sanz</i>  |           |
| <b>Patrones de adaptación para arquitecturas de software basadas en tecnologías del acuerdo</b>               | <b>6</b>  |
| <i>J. Santiago Pérez-Sotelo, Carlos E. Cuesta y Sascha Ossowski</i>   |           |
| <b>Análisis estático de código en el ciclo de desarrollo de software de seguridad crítica</b>                 | <b>26</b> |
| <i>Eduard Lluna</i>   |           |
| <b>Sección Actualidad Invitada:</b>   | <b>39</b> |
| <b>Perspectivas de la mejora de procesos de software</b>  |           |
| <i>José Antonio Calvo-Manzano, Cátedra Everis de Mejora de Procesos Software en el Espacio Iberoamericano</i> |           |

# **Patrones de adaptación para arquitecturas de software basadas en tecnologías del acuerdo**

J. Santiago Pérez-Sotelo<sup>1</sup>, Carlos E. Cuesta<sup>2</sup> y Sascha Ossowski<sup>1</sup>

<sup>1</sup> Centro para las Tecnologías Inteligentes de la Información y sus Aplicaciones (CETINIA)

<sup>2</sup> Grupo de Investigación VorTIC3, E.T.S. de Ingeniería Informática

Universidad Rey Juan Carlos - Madrid, España

{josesantiago.perez, carlos.cuesta, sascha.ossowski}@urjc.es

## **Resumen**

La creciente complejidad de los sistemas software requiere volver a concebir las estrategias de desarrollo y mantenimiento. Esto lleva a considerar a la auto-adaptación como una cuestión básica para la arquitectura. Simultáneamente, los sistemas multiagentes constituyen un enfoque genérico para resolver problemas complejos. Ciertas propuestas avanzadas usan organizaciones para proporcionar estructuras nuevas y más complejas. Sin embargo, todavía no proveen mecanismos para cambiar su composición o los tipos de elementos y, así, lograr una auténtica auto-adaptación. Este artículo propone una solución arquitectónica. El dinamismo necesario será proporcionado por un acuerdo emergente: una estructura arquitectónica evolutiva, basada en la combinación de controles y protocolos predefinidos. El objetivo es definir organizaciones adaptativas, con énfasis en los mecanismos (adaptativos) de coordinación. En este contexto, grupos de agentes orientados a servicios se reúnen, y pueden evolucionar, mediante patrones de adaptación hasta transformarse en una organización “estable”. En este artículo se presenta un ejemplo específico para demostrar el interés del enfoque y para debatir su aplicabilidad.

**Palabras clave:** Auto-adaptación, arquitecturas adaptativas, sistemas multi-agente, tecnologías

## **Adaptation patterns for software architectures based on agreement technologies**

### **Abstract**

The growing complexity of software systems is causing a re-conception of their development and maintenance strategies. Self-adaptation has been recently recognized as a basic architectural concern. Concurrently, multi-agent systems have been developed as a generic approach to solve complex problems and advanced approaches use organisations to provide further complex structuring. However, they still do not provide mechanisms to change their composition patterns and element types, to achieve real self-adaptivity. This work proposes an architectural solution: the required dynamism will be supported by an emergent agreement: an evolving architectural structure, based on combining predefined controls and protocols. The objective is to provide adaptive organisations, and the emphasis is in the coordination mechanism (also adaptive). Service-oriented agents gather together in this context and the resulting aggregate can evolve applying adaptation patterns. Eventually it would reach a “stable” agreement. A case study is also provided in order to show the interest of this approach, and discuss its applicability.

**Keywords:** self-adaptation, adaptive architecture, multi-agent Systems, agreement technologies, dynamism.

*Pérez-Sotelo J.S., Cuesta C.E. y Ossowski, S., “Patrones de adaptación para arquitecturas de software basadas en tecnologías del acuerdo”, REICIS, vol. 7, no.3, 2011, pp. 6-24. Recibido: 22-11-2011; revisado:1-12-2011; aceptado: 1-12-2011.*

## 1 Introducción

Los sistemas actuales son cada vez más complejos, y este hecho está llevando a los diseñadores de sistemas a replantear la estrategia para su gestión. Las dificultades van más allá de la administración de sistemas individuales; las tareas rutinarias que previamente eran gestionadas por humanos son ahora realizadas por los propios sistemas, incluso acciones relacionadas con sus propias funcionalidades. Los sistemas complejos pueden observarse a sí mismos y adaptar sus estructuras y comportamientos según sea necesario. Esto incluye no sólo al comportamiento funcional, sino también a las propiedades no-funcionales. Por lo tanto, si se considera la auto-adaptación como una cuestión básica a nivel de arquitectura, hay muchas características a tener en cuenta, según Hübner et al. [1]. Algunos de sus beneficios potenciales incluyen, por ejemplo, la escalabilidad en la construcción de sistemas-de-sistemas, o la abstracción para describir cambios dinámicamente, entre otros [2].

Al mismo tiempo, en el campo de la Inteligencia Artificial (IA), los Sistemas Multi-Agente (SMA) se han desarrollado como un enfoque genérico para resolver problemas complejos. Algunas de las propuestas avanzadas utilizan el concepto de *organizaciones* para proveer estructuras adicionales. Sin embargo, éstas aún pueden tener ciertas limitaciones para lograr una verdadera auto-adaptación, es decir: no disponer sólo de la capacidad de afectar su *configuración* sino también su propia *composición* y los tipos de los elementos que las forman.

La solución propuesta se basa en un *ecosistema de servicios* con Tecnologías del Acuerdo (AT – Agreement Technologies) [3][4][5], en el que se pretende proporcionar el dinamismo necesario. En este enfoque, los servicios se ofrecen a través de las *organizaciones* de agentes, independientemente de cómo éstos los exporten a su vez. El objetivo es proveer *organizaciones adaptativas*, haciendo énfasis en que los mecanismos de coordinación también habrán de ser adaptativos. El propósito es definir una “tecnología inteligente” genérica que pueda resolver problemas complejos, lista para ser utilizada en entornos *abiertos*, y capaz de *adaptarse* a futuras evoluciones.

Los agentes (orientados a servicios, en este caso) se reúnen en un contexto o entorno predefinido de *controles* y *protocolos*; el resultado de su unión puede evolucionar mediante la aplicación reglas definidas por ciertos *patrones de adaptación*, llegando eventualmente a un *acuerdo* “estable”.

Este artículo está estructurado como sigue: en la sección 2 se discute el núcleo de la propuesta, del que emerge una arquitectura adaptativa dentro de un contexto de SMA. También se presentan referencias a trabajos relacionados. La siguiente sección describe los conceptos que definen la estructura de estas tecnologías, la estructura del *acuerdo* en sí, que se define como transversal a cinco capas conceptuales y cuenta con la base de las llamadas Tecnologías del Acuerdo. La estructura del *acuerdo* también es incorporada en la definición de la plataforma THOMAS [6], que implementa sus conceptos y características, y que sirve como soporte a los desarrollos y experimentos adicionales. Asimismo se expone su evolución, ya que con sus características actuales la plataforma presenta ciertas limitaciones; por ejemplo, precisamente, en la adaptación a los cambios en el entorno, en lo relativo a la estructura de las organizaciones, en la aparición de nuevas normas y relaciones sociales entre agentes, etc. A continuación, se discuten conceptos y mecanismos que deben ser colocados en una capa por encima de THOMAS, capaces de generar las estructuras que definirán organizaciones emergentes, estructuras auto-organizadas y, en última instancia, *arquitecturas adaptativas*. En la sección siguiente se analizan los ya mencionados *patrones de adaptación*, incluyéndose una breve descripción de un patrón objeto de estudio y un ejemplo de protocolo especificado en *cálculo- $\pi$* . Finalmente se presentan conclusiones y futuras líneas de trabajo.

## **2 Hacia una arquitectura adaptativa**

Actualmente los SMA son cada vez más populares en IA para resolver problemas complejos con eficacia. Se han propuesto diversas estrategias de desarrollo para que sean flexibles y puedan coordinarse a efectos de adaptarse a los cambios en el entorno. Sin embargo, también se acepta que los SMA no tienen el éxito esperado en la industria del software [7] [8], y probablemente se deba a una cultura diferente de desarrollo. El enfoque de esta propuesta es salvar esta brecha utilizando *conceptos orientados a servicios*, los que sí cuentan con una mayor popularidad. Si, además, este enfoque demuestra la capacidad auto-adaptativa deseada, se podrá cumplir con la promesa original de los SMA, que el sistema sea capaz de adaptarse por sí mismo a las condiciones cambiantes del problema por resolver.

La arquitectura que dé soporte al modelo se define como *SMA abierto* y también como *orientada a servicios y centrada en organizaciones*. En las secciones siguientes se describirá brevemente el diseño básico de esta arquitectura, así como también su

evolución para que sea capaz de describir las estructuras emergentes. El objetivo es que este diseño describa una arquitectura verdaderamente auto-adaptativa.

## **2.1 Una arquitectura basada en agentes y orientada a servicios**

Debido a que se propone un entorno verdaderamente flexible y dinámico, se requiere el uso de capacidades semánticas y de alta tecnología. Por lo tanto, se considera el uso de agentes en un contexto más amplio, con el agregado de una capa superior de servicios para proveer una función de *interoperabilidad*. Es fácil concebir un servicio como una manera de presentar las capacidades operacionales de un agente, o mejor aún, de una colección de agentes como una *organización*, la que proveerá el *servicio*. El utilizar agentes permite, entre otras cuestiones, el tratamiento explícito de la semántica, una coordinación estructurada, su capacidad de aprendizaje, usar una metodología para el desarrollo de servicios y estructurarlos en organizaciones, etc.

Este trabajo tiene tres metas principales: la definición de una plataforma general para identificar la arquitectura subyacente *basada en agentes, orientada a servicios y centrada en organizaciones*, que conduzca a la plataforma fundamental para las Tecnologías del Acuerdo; segundo, la introducción de estructuras adicionales, para hacerla *adaptativa*; y tercero, la identificación de las estructuras genéricas de adaptación para las organizaciones, en la forma de una construcción del *acuerdo*, y su evolución.

La noción central es el *servicio*, el componente básico de la arquitectura es el *agente*, y la estructura que unifica los conceptos es la *organización*, concebida como una composición jerárquica y recursiva de agentes. Implícita en la definición de SMA se encuentra la necesidad de *registrar* a los agentes en el sistema, para separar aquellos que pertenecen a la arquitectura de los que no. Un enfoque similar será utilizado con los servicios. Para permitir su acceso externo, primero deberán ser explícitamente registrados y agrupados como parte de una entidad mayor – incluso un servicio “compuesto”. Estos servicios podrán ser descubiertos, posteriormente, por otras entidades, así como dentro del registro distribuido del sistema.

## **2.2 Trabajos relacionados: el rol de la coordinación**

Posiblemente sea mejor considerar el concepto de *coordinación* como una cuestión previa a la *adaptabilidad*. Cuando se utiliza un SMA como solución, el problema de la coordinación siempre está presente, y consecuentemente, la adaptabilidad también se ve comprometida. Un modelo de coordinación debería abarcar los temas de creación y destrucción de los agentes, su comunicación y su distribución espacial, así como la

sincronización y la distribución de sus acciones en el tiempo [9]. En un sistema de coordinación los componentes son: *entidades* (llamados también *coordinables*, cuyas interacciones se rigen por el modelo); *medios* (las abstracciones que rigen las interacciones); y *leyes* (que definen el comportamiento de los medios de coordinación en respuesta a la interacción) [9].

De acuerdo con [10], pueden tenerse en cuenta dos tipos de modelos de coordinación: los modelos *control-driven* (dirigidos por control) y los modelos *data-driven* (dirigidos por los datos). Los primeros se centran en el acto de la comunicación, mientras que los segundos lo hacen en la información intercambiada en ella. De hecho, en lugar de considerárselos modelos opuestos y diferentes, se podrían considerar como dos maneras diferentes de *observar* la coordinación.

Cuando se trata el caso específico de un meta-modelo de *espacio de tuplas* [9], tal vez el caso mejor conocido de coordinación *data-driven* (o generativa), las entidades basan sus interacciones (cooperación, competición, etc.) en las propias tuplas de datos. La coordinación, obviamente, tiene lugar mediante la producción, el consumo, etc. de las tuplas. En resumen, se crea un entorno de *comunicación generativa*.

Avanzando en la evolución de la coordinación, los *sistemas auto-organizados* tienen un creciente nivel de organización interna entre sus componentes en términos de interacciones, de su estructura, etc. [11][9]

Una reciente definición de *auto-organización de la coordinación* tomada de [12] plantea una gestión de interacciones del sistema con propiedades de auto-organización, donde las interacciones son locales y los efectos de coordinación global deseados aparecen por *emergencia*. Constructivamente, la auto-organización de la coordinación se consigue mediante un medio de coordinación distribuido sobre el entorno topológico, estableciendo reglas de coordinación *probabilísticas* y *dependientes del tiempo*.

Un enfoque significativo, y cada vez más popular, en el contexto de SMA ha sido considerar a los agentes dentro de *organizaciones*, y no de manera aislada. Sin embargo, esto también puede ser considerado como un enfoque de coordinación: en lugar de tener una coordinación normal de todo el sistema, las organizaciones permiten tener un *ámbito de coordinación*. En este sentido, hay una relación intrínseca entre la coordinación auto-organizada, que se mencionó anteriormente, y el enfoque de este artículo, basado en la definición de una arquitectura adaptativa en base a organizaciones *emergentes* - por lo que se obtendrá una *coordinación auto-organizada y con ámbito*.

### 3 Tecnologías del acuerdo: la plataforma y su evolución

La noción central en esta tecnología es el *acuerdo* entre entidades computacionales: *organizaciones*, en el nivel superior, y también *agentes*, en el nivel inferior. Se concibe como una construcción arquitectónica, que debe ser capaz de evolucionar para permitir la definición de un *acuerdo emergente* entre las entidades.

#### 3.1 Tecnologías del acuerdo

En este trabajo se utiliza el conjunto de tecnologías y enfoques conocidos globalmente como *Tecnologías del Acuerdo* [1]. Los aspectos considerados para proponer una coordinación basada en acuerdos se estructuran en forma de “torre” y por niveles, de modo que cada uno proporciona funcionalidad al nivel inmediatamente superior (ver Figura 1). De esta manera el acuerdo, por definición, está *estructurado en capas*. Intuitivamente puede verse que cuando se alcanza un acuerdo, los elementos situados en los niveles inferiores deben respetarlo en su propio nivel. Los agentes que forman una organización deben cumplir con los términos del acuerdo.

La "estructura en torre" define un conjunto de capas que expresan la esencia conceptual de un acuerdo. Las capas son:

Semántica. Es la capa inferior, ya que las cuestiones semánticas influyen sobre todas las otras. La alineación semántica de ontologías [13] debe tenerse en cuenta tanto para evitar las no correspondencias, como para tener un entendimiento común.

Normas. Esta capa tiene que ver con la definición de normas/reglas que determinan las limitaciones que los acuerdos, y el proceso para llegar a ellos, tienen que satisfacer. Pueden suponer roles estructurales que afectan (o controlan) el comportamiento de los agentes, entrelazados con el dominio semántico.

Organizaciones. Implican una súper-estructura que restringe la manera en que se consiguen los acuerdos mediante la fijación de la estructura social de los agentes, las capacidades de sus roles y las relaciones entre ellos. En definitiva, la arquitectura de un sistema multi-agente, estructurada en configuraciones semi-independientes.



Figura 1: estructura original de las Tecnologías del Acuerdo [1]

Argumentación y Negociación. Pueden ser vistos como los protocolos que definen la estructura de un acuerdo. Tiene que haber acuerdos a alto nivel (entre organizaciones) y también de bajo nivel (un pacto o acuerdo individual).

Confianza. Es la capa superior en la estructura. Se necesitan mecanismos de confianza que resuman la historia de los acuerdos y las ejecuciones subsecuentes de los mismos con el fin de construir relaciones a largo plazo [14].

Para mayores detalles acerca de estos conceptos, véase [4][5].

El enfoque propuesto provee los elementos requeridos para construir una arquitectura adaptativa, de modo que para definir un acuerdo emergente sólo se requiere la identificación de los patrones estructurales y el conjunto de protocolos inter-niveles. Pueden plantearse mayores refinamientos, por ejemplo considerar meta-elementos, o definir agentes específicos para realizar tareas de apoyo para el propio acuerdo (tales como sensores, observadores, controladores, planificadores, etc.), entre otros.

### 3.2 La plataforma

En esta subsección se presenta la arquitectura base para las tecnologías previamente discutidas, que fueron concebidas para ser soportadas por un SMA abierto.

Las actuales investigaciones en la plataforma se orientan a lograr una mayor capacidad y funcionalidad, aprovechando las características de los SMA. Es más, y desde este punto de vista, los servicios son específicamente utilizados para lograr una mayor interoperabilidad. La idea principal es exportar el *sistema de agentes* como un *sistema de servicios*. El ecosistema de servicios resultante será soportado tanto tecnológica como metodológicamente [15].

Estos conceptos se elaboran sobre la base de un sistema ya existente, la plataforma THOMAS [6]. Su diseño se resume en el gráfico siguiente (véase Figura 2).

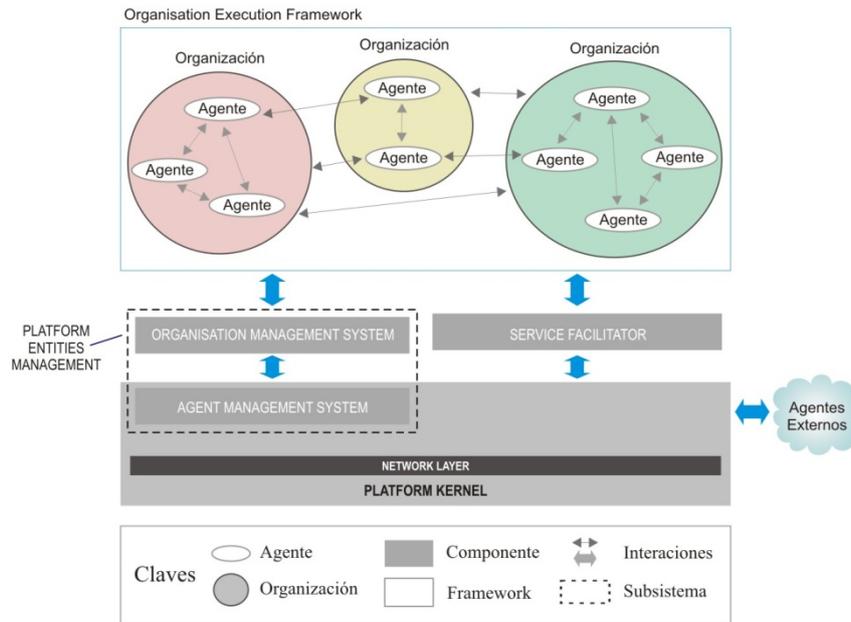


Figura 2: arquitectura técnica THOMAS (inspirado en [6])

La plataforma, incluido su *middleware*, está estructurada en tres niveles (aunque no son estrictamente capas) ortogonalmente soportados por cuatro componentes específicos que se incluyen como formando parte de tres diferentes subsistemas. A su vez, el subsistema *Platform Entities Management* está formado por capas, que se utilizan para proporcionar las capacidades a los diferentes niveles de la plataforma. Los tres niveles son:

- *Platform Kernel (PK)*. Es el *kernel* del middleware, incluye la capa de red y el componente Agent Management System (AMS). Provee las capacidades de una arquitectura compatible con FIPA [16]. A este nivel de descripción, la plataforma es un SMA abierto.
- *Service & Organisation Management*. Es el nivel conceptual compuesto por los componentes Organization Management System (OMS) y Service Facilitator (SF). No es estrictamente un subsistema, aunque provee las características y abstracciones relevantes para el Execution Framework.
- *Organisation Execution Framework*. Es el “espacio” donde las entidades computacionales se ejecutan y realizan sus tareas. Los agentes, sus organizaciones y los servicios que ofrecen están conceptualmente ubicados allí. Cada aplicación específica se concibe, diseña y ejecuta en este nivel de abstracción.

Los tres componentes principales mencionados de la plataforma son:

- *AMS*, que proporciona todas las capacidades y funciones necesarias para la gestión de los agentes;
- *OMS*, que hace lo propio con las organizaciones a la vez que mantiene unido al sistema en su conjunto;
- *SF*, que proporciona las capacidades y funciones necesarias para permitir que cierta selección de operaciones de una organización se comporte como un servicio unificado.

Más detalles acerca de la plataforma THOMAS pueden encontrarse en [6].

### 3.3 Evolución de la plataforma

Los progresos actuales sobre la plataforma se dan a nivel de los servicios con su adaptación a la especificación OSGi [17]. La idea principal es modularizar las aplicaciones en entidades más pequeñas denominadas *bundles*. Estas entidades pueden ser instaladas, actualizadas o eliminadas dinámicamente, proporcionando la posibilidad de cambiar el comportamiento del sistema sin tener que interrumpir su operación.

El *Service Tracker* se distingue entre los servicios que provee OSGi, especialmente para el enfoque propuesto: permite rastrear otros servicios registrados en la plataforma y también verificar que los que se han proveídos están o no disponibles. En un sistema basado en *bundles*, tanto productores como consumidores de servicios pueden aparecer y desaparecer dinámicamente. Para realizar el paso de mensajes, síncronos o asíncronos, entre dos entidades que pertenecen a distintos *bundles*, la especificación proporciona una herramienta: el patrón *Whiteboard*. Éste utiliza el registro de servicios de OSGi para mantener un registro de los *listeners* del sistema. Uno de los beneficios de esta herramienta es que el control del ciclo de vida de productores y consumidores de eventos se delega en la plataforma, que notifica a consumidores la desaparición de productores y viceversa.

Las actuales investigaciones, que se incluyen como parte del proyecto OVAMAH [18], se desarrollan ampliando los objetivos de la plataforma THOMAS. Además de proveer la tecnología necesaria para el desarrollo de organizaciones virtuales en entornos abiertos, permitirá dar *respuestas dinámicas* por medio de la adaptación y/o evolución de las propias organizaciones. Por ejemplo, agentes integrantes de una unidad organizacional podrán crear (o eliminar) otra unidad, afectando a los grupos del sistema; con respecto a las normas, podrán definir en qué momento es necesario agregarlas o eliminarlas; el tipo de relación social entre roles podrá cambiar en tiempo de ejecución,

tanto las condiciones para activarlas/desactivarlas como la cardinalidad entre roles; la topología del sistema (dada por las relaciones) podrá cambiarse (agregar/eliminar relaciones) también en tiempo de ejecución y luego verificarse que este cambio es consistente con los objetivos y el tipo organizacional; los servicios podrán ser asociados a nuevos roles que surjan en el sistema; etc.

#### **4 Definición del acuerdo emergente**

Cuando se utiliza un SMA abierto como solución de un problema complejo, a menudo se requiere cierta adaptación. Al mismo tiempo, la propia estructura debe ser flexible para lograr coordinación dentro del sistema.

El *acuerdo* entre entidades computacionales puede verse como un concepto adecuado para abordar la necesidad de una estructura adaptativa. El objetivo es descubrir una estructura apropiada, de modo que *emerja* como un acuerdo global. A continuación se definen conceptos y elementos que, contruidos sobre las estructuras básicas descritas en los apartados anteriores, harán posible definir una organización *emergente* sobre la base del *acuerdo*. Se pueden encontrar trabajos recientes relativos a las organizaciones dinámicas en [1][19][20].

##### **4.1 Canalizando la emergencia: controles y protocolos**

Cualesquiera que sean los objetivos, cualquier grupo de individuos puede ser ordenado según ciertas estructuras (por ejemplo una sociedad, una arquitectura, una jerarquía, etc.). Se puede provocar explícitamente la formación de estas estructuras mediante el uso de dos tipos diferentes de mecanismos, que tienen su fundamento en la limitación del rango de acciones válidas: *controles* y *protocolos*.

En cuanto a los primeros, los *controles*, pueden tanto *imponer* como *prohibir* interacciones específicas (a menudo, en la forma de conexiones arquitectónicas). Las estructuras auto-adaptativas, siendo típicamente centralizadas [11], muestran muchos ejemplos clásicos de este tipo: la mayoría de ellos manifiestan lazos explícitos de control, inspirados en los *reguladores* de la teoría de control clásica.

Por otro lado, los *protocolos*, que pueden *permitir* o *canalizar* comportamientos, se basan en el consenso y en acuerdos. Pueden ser descritos, genéricamente, como una manera de controlar las estructuras descentralizadas (o incluso distribuidas) [21]. Es decir, con los protocolos cada agente sabe cómo interactuar con el resto; es necesario

ponerse de acuerdo con ellos para poder comunicarse, y al mismo tiempo *regulan* el desarrollo de la estructura de interacción.

Estos dos mecanismos definen un amplio espectro de regulaciones. Las organizaciones de agentes y sus arquitecturas son simultáneamente regulados por controles unitarios, atómicos (como las *normas*, límites, *bloqueos*, lazos de control o *restricciones*), y protocolos múltiples, conectivos (como los *concentradores*, *puentes*, *canales* o *espacios*).

En arquitectura de software son conocidos muchos patrones y soluciones basados en controles implícitos y restricciones normativas. Sin embargo, el enfoque propuesto intenta establecer una solución basada en el *consenso*: también se trata de un enfoque a nivel arquitectónico, ya que la descripción de las interacciones pertenece a este nivel.

Es importante señalar que el propósito de estos mecanismos es “descubrir” una estructura adecuada de controles y protocolos para que pueda emerger una estructura global (es decir, la definición de diversas formas de arquitectura). Estos elementos harán posible definir las principales estructuras internas a fin de obtener organizaciones basadas en acuerdos.

Una vez que una estructura primaria pueda ser definida, un grupo elemental surge como una organización preliminar. Ésta será referenciada como una *iniciativa*, y su estructura será explicada en la siguiente subsección.

#### **4.2 Definición de un acuerdo emergente: la *Iniciativa***

Como ya se ha señalado, un conjunto de controles y protocolos puede ser utilizado para generar dinámicamente una organización preliminar dentro de un grupo de individuos (agentes, en este artículo, aunque también podrían ser componentes genéricos).

En esta propuesta se define y utiliza un conjunto de controles y protocolos para generar cierta estructura (por tanto, varios de ellos son considerados como *controles generativos* y *protocolos generativos*). Esta estructura da lugar a una organización que crece con la dinámica del entorno. La organización emergente es lo que llamaremos una *iniciativa*: todavía no está plenamente establecida, pero sigue evolucionando.

La *iniciativa* puede seguir creciendo y mutando debido a su naturaleza *adaptativa*, pero cuando se tiene algún tipo de estructura “estable”, ya es posible llamarla *organización*. Esta estructura se logra cuando todos los participantes pueden llegar al *acuerdo* necesario con el fin de solucionar el problema o conseguir el objetivo principal

que provocó su unión. La organización resultante es conceptualmente similar a otras organizaciones en varios enfoques de SMA, incluso en THOMAS [6].

El párrafo anterior implica tres conceptos importantes [22]:

- *Una iniciativa.* Es un grupo preliminar de individuos (agentes) que se reúnen en una determinada estructura, generada por un conjunto de controles y protocolos, así como también por ciertos patrones asociativos;
- *Una organización.* Se trata de un grupo establecido, dinámicamente originado desde una iniciativa (aunque también hay organizaciones “estáticas”, una vez que se crean, ambos tipos son funcionalmente equivalentes);
- *Un acuerdo.* Es el acto por el cual una iniciativa se convierte en una organización “estable”. De hecho, esto puede verse como el consenso que se alcanza entre los individuos dentro de la “semilla” inicial del grupo.

El proceso puede ser visto como si el sistema se moviera a un nuevo estado, en el que la estructura del “pasado” es suplantada por una “nueva estructura emergente”. Obviamente esta nueva estructura admite nuevos elementos debido al entorno dinámico, pero ahora uno de sus objetivos es reforzar su naturaleza y tender a la persistencia. Como las estructuras pueden ser cada vez más complejas, queda claro que para ciertas clases de problemas es necesario que los individuos se agrupen en organizaciones, y después, como ya se ha dicho, en organizaciones “estables” basadas en acuerdos.

Por ejemplo, en una situación de emergencia, varios coches policiales pueden llegar a un lugar determinado, pero ninguno de ellos es, a priori, el líder del grupo. Siguiendo un protocolo interno, pueden elegir a uno como tal, y este acuerdo genera una organización *preliminar*: incluso la jerarquía es un protocolo. Esto es lo que podría denominarse como un *protocolo generativo*; cuando los individuos siguen este tipo de protocolo, se definen patrones estructurales implícitos.

Una *iniciativa* puede generarse a partir de estos patrones, a los que llamaremos *patrones de adaptación* – téngase en cuenta que el término “patrones” es utilizado en un sentido arquitectónico. Son pre-diseñados a partir de los servicios necesarios por una iniciativa, y concebidos para el refinamiento semántico correspondiente. Algunos ya han sido identificados (véase Tabla 1): *Façade*, *Mediator*, *Surveyor*, entre otros.

### **4.3 Ciclo de vida de las estructuras auto-organizadas**

Como ya se ha dicho, un grupo de individuos puede organizarse, en base a metas concretas, en ciertas estructuras utilizando controles y protocolos. Estos mecanismos

hacen posible definir las estructuras internas relevantes para obtener organizaciones basadas en acuerdos. Una vez que la estructura primaria está definida, un grupo “elemental” emerge como entidad preliminar: la *iniciativa*. Ésta cambiará con la dinámica del entorno hasta ser una organización “estable”.

La Figura 3 resume brevemente el *ciclo de vida* de las estructuras auto-organizadas propuestas [23]. El ciclo puede comenzar con un simple *agente*, que está en condiciones de generar ciertas interacciones, y además tiene el potencial de exportar algunos de sus *servicios*. Inicialmente no pertenece a *organización* alguna cuando llega al sistema, sin embargo se ajusta a cierto número predefinido de *controles* y *protocolos*, que le guiarán en las interacciones y le permitirán mantener conversaciones estructuradas con otros agentes, formando grupos informales con ellos.

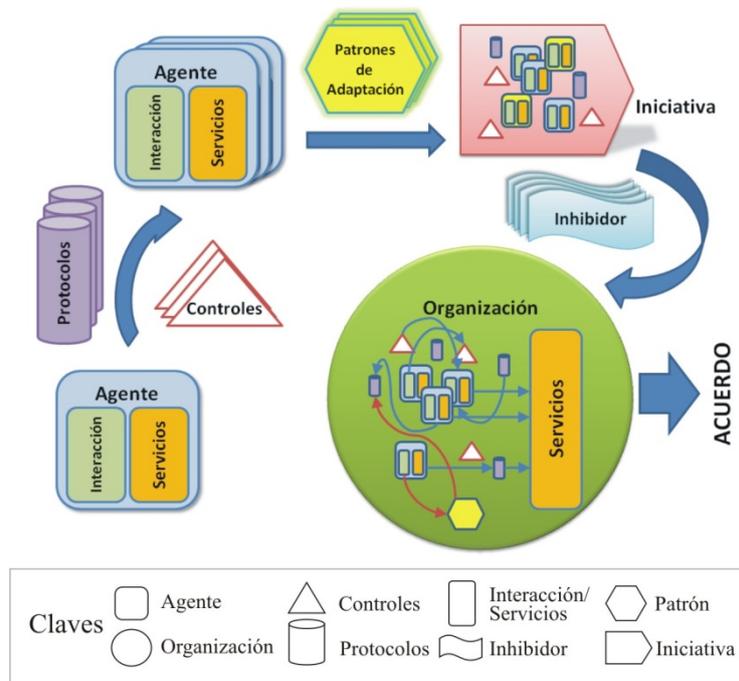


Figura 3: Ciclo de vida de una estructura auto-organizada [23]

Al ocurrir un cambio externo, el sistema debe reaccionar con un comportamiento *adaptativo*, y ésta es la funcionalidad que debe *disparar* la formación de las estructuras auto-organizadas (organizaciones). El sistema tiene a disposición cierto número de *patrones de adaptación* que pretenden, en conjunto, lograr la reacción deseada. Estos patrones son definiciones parciales de elementos y relaciones, y que incluyen la información suficiente para que un agente *aprenda* cómo realizar (o, siguiendo el protocolo, realice) cierto comportamiento. En resumen, guiados por un patrón de adaptación, ciertos agentes del grupo adquieren funciones específicas y comienzan a

forma una estructura: es lo que hemos llamado una *iniciativa*. Por supuesto, estas organizaciones pueden continuar evolucionando y participar, además, en otros acuerdos mayores [23].

## 5 Patrones de adaptación

Como ya se ha dicho, una *iniciativa* puede ser generada a partir de patrones, los llamados *patrones de adaptación*. Éstos son también patrones arquitectónicos: algunos de ellos ya han sido identificados, y reciben nombres como *Façade* o *Mediator*, entre otros [24].

El sistema está concebido como una *arquitectura orientada a servicios*, por lo tanto, metodológicamente, las primeras organizaciones “estables” deber ser concebidas como proveedoras de ciertos servicios de alto nivel. Estos servicios también serán usados como punto de partida para la definición funcional de las organizaciones evolutivas – incluso en su etapa inicial tratan de proporcionar esa funcionalidad.

| Nombre          | Descripción y funcionamiento.   |
|-----------------|---|
| <i>Façade</i>   | Algún agente ha de representar a la propia organización para poder interactuar fácilmente con una que aún carece de estructura definida. Este agente será la <i>fachada</i> que redirige cualquier comunicación recibida; no es necesario que sea también un supervisor.  |
| <i>Mediator</i> | Durante el proceso de emergencia, la organización aún no está establecida y los servicios de datos posiblemente no estén funcionando. Algún agente debe actuar como mediador haciendo posible el acceso a fuentes de datos, al menos indirectamente, y también realizar las traducciones necesarias, incluyéndose distintas clases de traducción <i>semántica</i> . |
| <i>Surveyor</i> | Al menos un agente debe controlar el crecimiento de la propia iniciativa durante el proceso de emergencia. Asimismo puede decidir el momento de introducir nuevos elementos y también cuándo se estabiliza el grupo. El <i>surveyor</i> tiene acceso a la biblioteca de patrones – decide cuándo cierto patrón es el adecuado, y debe ser activado.                 |

Tabla 1: Algunos patrones de adaptación: patrones de diseño a nivel arquitectónico

Como ejemplo sencillo se ha seleccionado otro patrón de adaptación: *Gathering*. A continuación, y de acuerdo con [25], se proporciona un extracto de la plantilla de este patrón (véase Figura 4). La plantilla es similar en estilo y espíritu a los usados por [26], pero algunos campos han sido modificados para denotar la dinámica de los sistemas adaptativos. Téngase en cuenta que varios campos, y sus descripciones, se han omitido o simplificado debido a restricciones de espacio.

El patrón describe la situación donde varios agentes se encuentran cerca de un punto de encuentro (*venue*). Cuando ocurre un evento en este punto, los agentes se encuentran capacitados para interactuar directamente entre sí: en resumen, este evento inicial puede disparar la creación de una *iniciativa*.

Como ocurre en otros patrones, la plantilla debe complementarse con algún modelo de interacción – en los patrones de adaptación los *protocolos* son de particular importancia: esta es la razón por la que se usará una especificación en *cálculo- $\pi$*  [27]. Se utiliza este cálculo debido a su gran expresividad; pero no es necesario utilizarlo para la codificación de los patrones: de hecho, se podrían utilizar otros mecanismos (o incluso formalismos) a estos efectos.

Esta especificación esencialmente utiliza el cálculo- $\pi$  poliádico estándar, con algunas y convenientes alteraciones. Se utiliza la notación CSP en lugar de la original, por lo que también se reserva el símbolo asterisco (\*) para representar la operación de *replicación*. También se asume una construcción (;) para una composición secuencial, y se incluye además el condicional (*if A then B*). Ambos se codifican convencionalmente de una manera estándar [27]. Se agregan también dos funciones simples, fácilmente implementables, por conveniencia algorítmica: **near** (detecta proximidad) y **count**. En la Figura 5 puede leerse un extracto de la especificación.

- Nombre del Patrón. **Gathering (Reunión)**
- Clasificación. Monitorización, Creacional.
- Propósito. Monitorizar el “espacio” que contiene elementos computacionales (agentes), canalizar los datos observados, facilitar la interacción y, en última instancia, proporcionar coordinación.
- Contexto. [*Describe el contexto de aplicación del patrón.*] *Gathering* puede usarse cuando aún no se ha formado una *iniciativa*, es decir, cuando los agentes pueblan el “espacio” pero todavía no se han definido los servicios de alto nivel, proporcionando una organización.
- Motivación. El sistema necesita coordinarse como un ecosistema de servicios, en el que los agentes puedan entrar o salir del entorno dinámicamente. Esto justifica un comportamiento que no podría

Figura 4: Plantilla del patrón de adaptación Gathering (extracto)

```

AgentX (sys_where) ::= ...
    + (v hi) sys_where!⟨hi⟩. sys_where?(where). τ. where?(event).
      (v me, ret) * [ event!⟨me, ret⟩. ret?(you). τ. (me!⟨data⟩ |
you!⟨info⟩) ];
      AgentX⟨sys_where⟩
    + ...

Venue (sys_place) ::= ...
    + (v here, r) sys_place!⟨here, r⟩. τ.
      * [ r?(hi). if count(r) ≥ 2 then (v event) here!⟨event⟩.
        * [ event?(ag_x, ragx). (
          * [ event?(ag_y, ragy). ragx!⟨ag_y⟩ ] | event!⟨ag_x, ragx⟩
        ) ] ];
      Venue ⟨sys_place⟩
    + ...

SubSystem (sys_place sys_where) ::=

```

Figura 5: Extracto de especificación en cálculo- $\pi$  para agentes cerca del punto de encuentro (venue)

El proceso *AgentX* describe la manera en que un agente estándar debe comportarse en el patrón. En primer lugar, es consciente de su existencia (*hi*) y pregunta al sistema dónde se encuentra (*sys\_where*). Si está cerca de un punto de encuentro (*venue*) se le envía su nombre (*where*). Cuando algo (*event*) sucede allí, el agente envía su propio nombre (*me*) y un canal de respuesta (*ret*). Por este canal, recibe el nombre de otro agente (*you*). Por lo tanto, los dos agentes pueden ahora interactuar directamente, y no volver a usar el *venue*. Ésta última parte es un sub-proceso repetitivo ( $*[A]$ ), que se repite indefinida y concurrentemente – lo que significa que el agente puede comunicarse con muchos otros, dos a dos.

El proceso *Venue* describe la influencia del “punto de encuentro” en los agentes. Primero informa al sistema de su existencia (*here*), y provee un canal de respuesta (*r*), en el que los agentes eventualmente podrán registrarse. Una vez que haya dos o más agentes, el evento se dispara y se retransmite dentro de un proceso repetitivo. El canal *event* es utilizado para recibir el nombre (*ag\_x*) y devolver el canal a todos los agentes cercanos, encontrar una pareja potencial (*ag\_y*) y comunicarse con ella, dentro de un proceso paralelo. Por supuesto hay otras maneras más sofisticadas para definir un protocolo de difusión (*broadcast*): éste es solo un ejemplo. El proceso *SubSystem* obviamente actúa como conector, esto es, define el comportamiento de la porción de *middleware* que aglutina al conjunto.

Como comentario final, puede verse que cada proceso AgentX es un *shifter* [23]: esto es, un “agente que cambia” – está cambiando su forma, es decir, su capacidad para interactuar con otros. En el mismo contexto, el *Venue* se define como un *changent*: esto es, un “agente de cambio” – que es el agente que *induce el cambio* sobre el resto de los elementos del patrón.

## **6 Conclusiones y trabajos futuros**

Este trabajo ha explorado conceptos estructurales como base de un enfoque arquitectónico para proporcionar auto-adaptación a sistemas software. El concepto de *iniciativa* puede ser considerado como un punto de partida para proveer mecanismos que permitan el cambio de los patrones de composición y de los tipos de elementos dentro de tales sistemas.

El dinamismo necesario puede ser soportado por un acuerdo emergente – una estructura arquitectónica evolutiva, basada en la combinación predefinida de controles y protocolos: éstos son gestionados en el contexto del marco orientado a servicios, basado en agentes y centrado en organizaciones definidos en las *Tecnologías del Acuerdo*, provisto por la implementación en la plataforma THOMAS, el proyecto OVAMAH y servicios compatibles con el estándar OSGi.

La idea clave es crear un contexto arquitectónico en el que los agentes son *coordinados y reorganizados* por su inclusión en estructuras preliminares –los *patrones de adaptación*– y luego compuestos en organizaciones “estables”.

Los patrones propuestos pueden ser entendidos como las estructuras necesarias para lograr una auto-adaptación real. Los protocolos que definen los patrones se pueden representar formalmente, como se puede ver en la especificación en cálculo- $\pi$  incluida. De hecho, aún cuando el enfoque propuesto parece prometedor, estos son únicamente los primeros pasos: el potencial de crecimiento es significativo.

En trabajos futuros se desarrollarán e implementarán variantes de este enfoque en el marco de una nueva plataforma de agentes basada en THOMAS (denominada AT-MAS) a fin de lograr su refinamiento. Los conceptos todavía están evolucionando, pero aún en esta fase los fragmentos del enfoque ya han demostrado su utilidad, y su poder expresivo. Los resultados sugieren que la meta final, esto es, definir una auténtica *arquitectura adaptativa*, es viable, dado que la infraestructura que se ha creado puede crecer simplemente añadiendo nuevos patrones de adaptación.

## **Agradecimientos**

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación de España, para los proyectos “Tecnologías del Acuerdo” (CONSOLIDER CSD2007-0022, INGENIO 2010), “OVAMAH” (TIN2009-13839-C03-02) y “MULTIPLE” (TIN2009-13838); y el “European Union RTD Framework Programme”, a través de la Acción COST Agreement Technologies (COST Action IC0801).

## **Referencias**

- [1] Hübner, J.F., Boissier, O., Kitio, R., y Ricci, A., “Instrumenting multi-agent organisations with organisational artifacts and agents: Giving the organisational power back to the agents”, *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 20, nº 3, pp. 369-400, 2010.
- [2] Kramer, J. y Magee, J., “Self-Managed Systems: an Architectural Challenge”, *Future of Software Engineering (FOSE@ ICSE'2007)*, pp. 259-268, IEEE, 2007.
- [3] Agreement Technologies (AT) Project, [www.agreement-technologies.org](http://www.agreement-technologies.org) (último acceso 23/01/2012).
- [4] Ossowski, S., “Coordination in Multi-Agent Systems: Towards a Technology of Agreement”, *MATES, LNCS 5244*, pp. 2-12, Springer, 2008.
- [5] Sierra, C., Botti, V.J., y Ossowski, S., “Agreement Computing”, *KI*, vol. 25, nº 1, pp. 57-61, 2011.
- [6] Argente, E., Botti, V., Carrascosa, C., Giret, A., Julian, V., y Rebollo, M., *An Abstract Architecture for Virtual Organisations: The THOMAS Project*, Technical report, DSIC, Universidad Politécnica de Valencia, 2008.
- [7] DeLoach, S., “Moving multi-agent systems from research to practice”, *International Journal of Agent-Oriented Software Engineering*, vol. 3, nº 4, pp. 378-382, 2009.
- [8] Weyns, D., Helleboogh, A., y Holvoet, T., “How to get multi-agent systems accepted in industry?”, *International Journal of Agent-Oriented Software Engineering*, vol. 3, nº 4, pp. 383-390, 2009.
- [9] Ciancarini, P., “Coordination models and languages as software integrators”, *ACM Computing Surveys*, vol. 28, nº 2, pp. 300-302, 1996.
- [10] Papadopoulos, G.A., y Arbab, F., “Coordination models and languages”. En Zelkowitz, M.V. (Ed.), *The Engineering of Large Systems*, vol. 46, pp. 329-400. Academic Press, 1998.

- [11] Andersson, J., de Lemos, R., Malek, S., y Weyns, D., “Modelling Dimensions of Self-Adaptive Software Systems”, *Software Engineering for Self-Adaptive Systems*, LNCS 5525, pp. 27-47, Springer, 2009.
- [12] Casadei, M., y Viroli, M., “Applying self-organising coordination to the emergent tuple organisation in distributed networks”. En Brueckner, S. et al. (Eds.), *2nd IEEE International Conference on Self-Adaptive and Self-Organising Systems – SASO*, 2008.
- [13] Atienza, M., Schorlemmer, M., “I-SSA - Interaction-situated Semantic Alignment”, *Proceedings International Conference on Cooperative Information Systems*, 2008.
- [14] Sierra, C., Debenham, J., “Information-Based Agency”, *Proceedings International Joint Conferences on AI (IJCAI-2007)*, pp. 1513-1518, AAAI Press, 2007.
- [15] Pérez-Sotelo, J.S., Cuesta C.E., y Ossowski S., “Agreement Technologies for Adaptive, Service-Oriented Multi-Agent Systems”, *Proceedings II Workshop on Agreement Technologies (WAT 2009) – XIII Conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA2009*, vol. 635, CEUR Workshop Proceedings, 2009.
- [16] FIPA Abstract Architecture Specification. *Technical Report SC00001L, Foundation for Intelligent Physical Agents*, FIPA TC Architecture, 2002.
- [17] OSGi Alliance, [www.osgi.org](http://www.osgi.org) (último acceso 23/01/2012)
- [18] OVAMAH - Organizaciones Virtuales Adaptativas: Técnicas y Mecanismos de Descripción y Adaptación, [gti-ia.dsic.upv.es:8080/ovamah](http://gti-ia.dsic.upv.es:8080/ovamah) (último acceso 23/01/2012)
- [19] Kolp, M., Giorgini, P., y Mylopoulos, J., “Multi-Agent Architectures as Organizational Structures”, *Autonomous Agents and Multi-Agent Systems (AAMAS)*, vol. 13, nº 1, pp. 3-25, 2006.
- [20] Weyns, D., Haesevoets, R., Helleboogh, A., Holvoet, T., y Joosen W., “The MACODO middleware for context-driven dynamic agent organizations”, *Journal ACM Transactions on Autonomous and Adaptive Systems*, vol. 5, nº 1, pp. 1-28, 2010.
- [21] Galloway, A.R., *Protocol: How Control Exists after Decentralization*, MIT Press, 2004.
- [22] Pérez-Sotelo, J.S., Cuesta C.E., y Ossowski S., “Towards Adaptive Service Ecosystems with Agreement Technologies”, *Proceedings I Workshop Adaptation in*

*serVice EcosYsTems and ArchiTectures (AVYTAT2010) - OnTheMove Federated Conferences 2010*, LNCS 6428, pp. 77-87, Springer, 2010.

- [23] Cuesta, C.E., Pérez-Sotelo, J. S., y Ossowski, S., “Self-Organising Adaptive Structures: the Shifter Experience”, *European Research Consortium for Informatics and Mathematics - ERCIM News*, nº 85, pp. 35-36, 2011.
- [24] Billhardt, H., Centeno R., Fernandez A., Hermoso R., Ortiz R., Ossowski S., Perez-Sotelo J.S., Vasirani M., y Cuesta C.E., “Organisational Structures in Next-Generation Distributed Systems: Towards a Technology of Agreement”, *Multiagent and Grid Systems: An International Journal*, vol. 7, nº 2-3, pp. 109–125, 2011.
- [25] Ramírez, A. J., y Cheng, B. H. C., “Design Patterns for Developing Dynamically Adaptive Systems”, *Proceedings ICSE2010-SEAMS*, pp. 49-58, 2010.
- [26] Gamma, E., Helm, R., Johnson, R., y Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994.
- [27] Sangiorgi, D., Walker, D., “The Pi-calculus: a Theory of Mobile Processes”. *Cambridge University Press*, 2003.