

# El Papel de las Tecnologías del Acuerdo en la Definición de Arquitecturas de Software Adaptativas

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

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

<sup>2</sup> Kybele Grupo de Investigación, Depto. de Lenguajes y Sistemas Informáticos II

Universidad Rey Juan Carlos,

28933 Móstoles (Madrid), España

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

**Abstract.** La creciente complejidad de los sistemas de software está originando que las estrategias de desarrollo y mantenimiento sean re-concebidos. Una tendencia es que gran parte de estas tareas deberían ser realizadas por los propios sistemas relevando de ellas al ser humano. Esto lleva a considerar a la auto-adaptación de sistemas como una cuestión básica de arquitectura. Al mismo tiempo, los sistemas multiagentes se han desarrollado como un enfoque genérico para resolver problemas complejos. En ellos se describen estructuras auto-conscientes, son concebidos para ser flexibles y capaces de adaptarse a diferentes situaciones. Ciertos enfoques avanzados recurren al concepto de organizaciones para proveer nuevas estructuras, tomando la forma de arquitecturas complejas. Entre ellos, las *Tecnologías del Acuerdo* proporcionan una visión explícita de esas abstracciones arquitectónicas. Sin embargo, aún no proveen mecanismos para cambiar sus patrones de composición y tipos de elementos, necesarios para lograr una auto-adaptabilidad real. En este trabajo se propone una solución arquitectónica: el dinamismo requerido será soportado por un acuerdo emergente – una estructura arquitectónica evolutiva, basada en la combinación de controles y protocolos predefinidos. Estos son tratados en el contexto de un marco orientado a servicios, basado en agentes y centrado en organizaciones, definido en las *Tecnologías del Acuerdo* y provisto por la implementación en la plataforma THOMAS. El trabajo proporciona las primeras abstracciones arquitectónicas para soportar esta estructura emergente. Se presenta también un ejemplo del mundo real para demostrar la utilidad de este enfoque y finalmente se tienen algunas conclusiones acerca de su aplicabilidad.

**Palabras clave:** Auto-adaptación, Arquitectura Adaptativa, Sistemas Multi-Agente, Tecnologías del Acuerdo, Arquitectura Dinámica.

## 1 Introducción

Es sabido que en los últimos años los sistemas de software han tenido un gran crecimiento en complejidad. Este nivel de complejidad, que puede ser llamada “social” según [14], está obligando a los diseñadores de software a repensar las estrategias para su estudio. Muchas de las tareas rutinarias que previamente estaban reservadas a usuarios humanos ahora están siendo manejadas por los propios sistemas, incluso acciones relacionadas con las funciones propias del sistema. Los sistemas complejos pueden ahora

observarse a sí mismos, y adaptar su estructura y comportamiento si fuera necesario. Por tanto, como este enfoque tiene una influencia global en el sistema y a muchos niveles [18], se puede considerar la *auto-adaptación* como una cuestión básica de la arquitectura [19].

Al mismo tiempo, los Sistemas Multi-Agente (SMA) se han desarrollado como un enfoque genérico en Inteligencia Artificial (IA) para resolver problemas complejos. Describen estructuras auto-conscientes, pueden tener capacidad de aprendizaje, son diseñados para ser flexibles y capaces de adaptarse a diferentes situaciones, etc. Ciertos enfoques avanzados utilizan el concepto de organizaciones para proveer nuevas estructuras, las que originan arquitecturas complejas. Sin embargo, las estructuras existentes aún pueden tener limitaciones al intentar alcanzar una auto-adaptación real, es decir, no sólo tener la capacidad de cambiar su configuración, sino también su propia composición o tipos de elementos constituyentes. En este trabajo se intenta ir más allá de las tecnologías “clásicas” de agentes y proponer una solución basada en las *Tecnologías del Acuerdo* (Agreement Technologies) [1][24] para enfrentar el dinamismo.

Este trabajo está organizado de la siguiente manera: en la segunda sección se presenta un ejemplo motivador para ilustrar las ideas principales y el enfoque propuesto, definido como orientado a servicios, centrado en organizaciones y basado en agentes. En la sección siguiente 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 relativos. La cuarta sección presenta los conceptos que apoyan 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 *Tecnologías del Acuerdo* [24]. La estructura del *acuerdo* también es incorporada en la definición de la plataforma THOMAS [3], que implementa sus conceptos y características, y que sirve como soporte a los desarrollos y experimentos adicionales. Posteriormente se discuten los conceptos y mecanismos que deben ser colocados en una capa por encima de THOMAS, capaz de generar las estructuras que definirán organizaciones emergentes y, en última instancia, las arquitecturas adaptativas. Seguidamente, el ejemplo motivador es examinado en el contexto de THOMAS, dando una idea de la forma en que se aplican las características adaptativas. Por último se presentan algunas conclusiones y futuras líneas de trabajo.

## 2 Ejemplo Motivador

Con el fin de ilustrar la situación en la que una arquitectura adaptativa puede ser tomada como solución, en esta sección se presenta un ejemplo motivador describiendo un escenario en el dominio de las emergencias médicas. El ejemplo se relaciona con el módulo demostrador *mHealth* (mobile-Health), un prototipo actualmente en desarrollo en el proyecto *Tecnologías del Acuerdo* [1] con la colaboración del servicio “SUMMA112” [30]. Esta última gestiona las emergencias médicas en la Comunidad Autónoma de Madrid, España. En [8] se puede encontrar información sobre apoyo a emergencias médicas mediante el uso de un SMA estándar.

El ejemplo propuesto describe una emergencia inicial (E1) en el sistema, el que luego tiene que evolucionar para reaccionar de forma simultánea a una segunda (E2). Esta situación muestra que la adaptación no solo requiere la reorganización de los agentes, sino también nuevas configuraciones del sistema y hacer frente a nuevos problemas de coordinación.

**E1.** Se produce un incendio en un gran parque urbano situado al oeste del centro de Madrid, el que contiene una amplia zona de ocio. Alrededor de 500 personas se encuentran en el parque en ese momento, de las cuales 65 aproximadamente presentan síntomas de asfixia, además el fuego se está extendiendo a las zonas adyacentes con celeridad. El SUMMA112 recibe la información correspondiente y decide que son necesarios 5 ambulancias y un helicóptero. Es también urgente realizar la coordinación con los departamentos Bomberos y Policía. Estos últimos informan que enviarán 3 camiones bomberos y 5 coches policiales, respectivamente. Desde un enfoque organizacional, todos estos elementos forman una organización: O1. Teniendo en cuenta esta situación en un entorno SMA, cada uno de estos actores se mapea en un agente. Visto así se tienen 14 agentes interactuando en la organización O1. Como es obvio, cada agente tiene su función, objetivos y planes dentro de la organización. Asimismo, cada organización tiene sus normas y protocolos, que le permite funcionar y operar de manera eficiente para resolver las situaciones de emergencia.

**E2.** Una hora después de E1 se produce un accidente automovilístico en cadena (E2) en un túnel de una carretera que se encuentra cerca de E1: 7 coches se han estrellado y dos de ellos están con principio de incendio. SUMMA112 decide que esta emergencia requiere 3 ambulancias y que debe ponerse en contacto con los hospitales cercanos de la zona. Al coordinar con Bomberos y Policía, éstos deciden enviar un camión de bomberos y tres coches policiales. De nuevo, todos estos elementos (inicialmente 7) forman una segunda organización: O2.

Pueden considerarse dos soluciones alternativas para resolver este escenario:

- hacer frente a O1 y O2 como elementos separados (unidades organizativas separadas), sin relación entre ellos, o
- tratar a O1 y O2 como unidades con algún grado de relación entre ellas.

La segunda se corresponde con un enfoque más eficiente y razonable, al tener en cuenta las posibles interacciones entre ambas emergencias. En el momento de asignar recursos a E2, O2 no se considera en forma aislada de O1. Algunos recursos que antes fueron asignados a O1 ahora podrían ser asignados a O2. Esta situación es factible, ya que las condiciones de E1 pueden haber cambiado durante la última hora. Este proceso de re-asignación implica una reconfiguración de la unidad O1, es decir, la reorganización de un agente dentro del compuesto O1O2. Metodológicamente, algunos servicios que fueron proporcionados por la unidad O1 ya no son necesarios para resolver E1 y son ahora re-asignados en O2.

Teniendo en cuenta los resultados en O1 (o utilizando algún tipo de métrica), el sistema decide, por ejemplo, que uno de los camiones bomberos ya no es necesario en E1 y puede ser asignado a E2. La misma decisión se puede tomar con 2 ambulancias y 2 vehículos policiales. Estas re-asignaciones también son eficientes porque ambos escenarios están cercanos.

Esto podría ser visto también como si O1 se dividiera en dos organizaciones: una sigue su labor en E1, mientras que la otra comienza su trabajo en E2. Por supuesto que otros elementos también deben ser asignados a E2: se necesitan una ambulancia y un coche policial más. El original O1, con un conjunto más pequeño de elementos, sigue trabajando en E1, y un nuevo *acuerdo* se crea alrededor de E2, definiéndose la organización O2. Al mismo tiempo se crea un *acuerdo* más amplio abarcando ambas unidades organizacionales. Todo el sistema seguirá adaptándose a los cambios que ocurran en ambas emergencias, incluso reasignando elementos en caso de ser necesario.

El escenario descrito es clara y necesariamente adaptativo. Como el entorno está orientado a agentes el objetivo es lograr una reconfiguración automática: el sistema debe llevar a cabo una serie de pasos evolutivos hasta que encuentre un punto óptimo. Esto bien puede ser un proceso continuo, ya que el propio escenario va evolucionando.

Con este ejemplo queda claro que este comportamiento podría no ser completamente pre-diseñado sino que debería ser emergente y lograrse la coordinación dentro de una arquitectura que es, básicamente, un ecosistema de servicios (es decir, un conjunto de servicios que fueron creados separadamente pero que deben interactuar y coordinarse en un determinado contexto). En la sección siguiente se presentan conceptos y herramientas desarrollados con el fin de definir las bases de una solución y como validación del enfoque propuesto.

### 3 Hacia una Arquitectura Adaptativa de Agentes

El concepto de *agente* ha evolucionado y hoy día los SMA son populares en IA como una manera eficaz para resolver problemas complejos. Se han propuesto diferentes estrategias de desarrollo con el fin de que sean flexibles y puedan coordinarse para adaptarse a entornos cambiantes. Sin embargo, también se acepta que los SMA no han tenido todo el éxito esperado en la industria del software [12][31], y esto debido, probablemente, a una cultura diferente de desarrollo.

El enfoque que se propone es salvar esta brecha utilizando *conceptos orientados a servicios*, los que sí cuentan con una mayor popularidad. Por otra parte, si este enfoque demuestra también la capacidad auto-adaptativa deseada, cumplirá con la promesa original de los SMA: garantizar que el sistema sea capaz de adaptarse a condiciones cambiantes del problema a resolver.

La arquitectura que da soporte al modelo se ha definido como un *SMA abierto*, así como también *orientada a servicios, centrada en organizaciones y basada en agentes*. A continuación se describe brevemente el diseño básico definido para esta arquitectura y posteriormente se la extiende para que sea capaz de describir las estructuras emergentes. El objetivo es que el diseño describa una arquitectura verdaderamente auto-adaptativa.

#### 3.1 Una Arquitectura Orientada a Servicios y Basada en Agentes

Como fue señalado, el enfoque propuesto se basa en conceptos orientados a servicios. Aunque la tecnología de servicios (en particular SOA) está establecida y ya tiene una serie de estándares [6][9][13][20], su metodología y su influencia en otros paradigmas (p. ej. las arquitecturas orientadas a agentes) todavía está en desarrollo. La función *descubrimiento de servicios* proporciona cierta flexibilidad a una arquitectura orientada a servicios, pero estas arquitecturas están fuertemente ligadas por sus semánticas y coreografías. Se pueden considerar como excepciones a los *mashups* (o híbridos de aplicaciones web), pero aún así suponen soluciones ad-hoc [21].

Dado que el entorno propuesto debe ser verdaderamente flexible y dinámico, se requiere el uso de amplias capacidades semánticas y de alta tecnología. Por lo tanto, se considera un uso racional de agentes en un contexto más amplio, con el agregado de una capa superior de servicios para proveer una función de *interoperabilidad*. El utilizar agentes permite el tratamiento explícito de la *semántica*, una *coordinación estructurada*, el uso de una

metodología para el desarrollo de servicios y estructurarlos en organizaciones, el uso de su capacidad de *aprendizaje*, entre otras cuestiones.

En este punto podemos proponer una agenda de investigación en tres fases:

1. 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*;
2. La introducción de estructuras adicionales, para hacerla *adaptativa*;
3. 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 recursiva de agentes.

Implícita en la definición de un SMA está la necesidad de *registrar* los agentes del sistema para identificar aquellos que pertenecen a la arquitectura. El mismo criterio se utilizará con los servicios. Para permitir el acceso externo, serán explícitamente registrados y agrupados como parte de uno “mayor”. Más tarde, este servicio puede ser descubierto por otras entidades en el registro distribuido del sistema.

### 3.2 Trabajos Relacionados: el Rol de la Coordinación

Tal vez sea mejor considerar la *coordinación* como cuestión previa a la *adaptabilidad*. Desde un punto de vista centrado en SMA, la coordinación puede ser vista como un plan “compartido” [28] o la combinación de planes individuales (un “multi-plan”) [23]. Un modelo de coordinación debería abarcar los temas de creación y destrucción de agentes, su comunicación, y la distribución espacial entre ellos, así como la sincronización y la distribución de sus acciones en el tiempo [10].

En un sistema de coordinación los componentes son: *entidades* (también llamados *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) [10].

De acuerdo con [25] pueden tenerse en cuenta dos tipos de modelos de coordinación: los modelos *control-driven* versus los *data-driven*. Los primeros se enfocan en el acto de la comunicación y los segundos en la información intercambiada durante ella. Se podrían considerar como dos maneras diferentes de *observar* la coordinación.

Cuando se trata de un meta-modelo de espacio de tuplas [10][23], las entidades basan sus interacciones (cooperación, competición, etc.) en tuplas. La coordinación tiene lugar en un espacio mediante producción, consumo, etc. de tuplas. En resumen, se crea por la *comunicación generativa*. Si se avanza en la evolución de la coordinación, los *sistemas auto-organizados* tienen un creciente nivel de organización interna entre sus componentes en términos de interacciones, su estructura, etc. [2][7]

Una reciente definición de la *auto-organización de la coordinación* tomada de [7] 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*.

Uno de los enfoques más significativos en el contexto de SMA ha sido considerar agentes dentro de *organizaciones* y no en aislamiento [4]. Sin embargo, esto puede ser considerado en definitiva 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, como se mencionó anteriormente, y el enfoque de este trabajo, basado en la definición de una arquitectura adaptativa en base a organizaciones *emergentes* - por lo tanto, se tendrá una *coordinación auto-organizada y con ámbitos*.

## 4 Tecnologías del Acuerdo: la Plataforma THOMAS

La noción central en la propuesta 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.

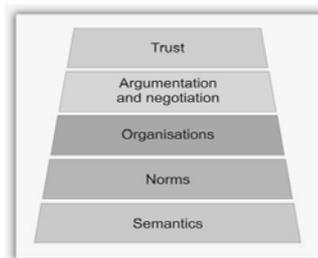
### 4.1 Tecnologías del Acuerdo

En este trabajo se utiliza el conjunto de tecnologías y enfoques conocidos globalmente como *Tecnologías del Acuerdo* [1]. Los tópicos a considerar para proponer una coordinación basada en acuerdos pueden ser vistos como estructurados en “torre”. Cada nivel de la torre proporciona funcionalidad y aportes a la de arriba, (Figura 1), es así que el acuerdo se estructura en capas, por definición. 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. Las capas son las siguientes:

Semántica. Es la capa inferior, ya que las cuestiones semánticas influyen sobre todas las otras. La alineación semántica de ontologías [5] debe tenerse en cuenta para evitar las no correspondencias y 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 super-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 [4].



**Figura 1:** Estructura original (en capas) de la torre 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 de la torre. 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 entre ellos [29].

En [24] pueden encontrarse más detalles acerca de los conceptos claves mencionados.

Las cinco capas pueden beneficiarse entre sí; de hecho, el acuerdo es una estructura transversal, que mantiene una relación bidireccional entre todos los elementos que contiene [26]. El acuerdo está moldeado por las fuerzas del entorno y su existencia conforma la reacción a ellas y modela la evolución futura del sistema.

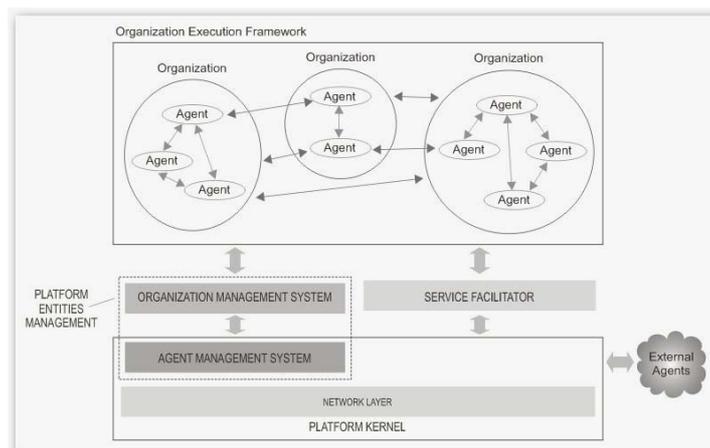
En resumen, para definir un acuerdo emergente sólo se requiere la identificación de los patrones estructurales y el conjunto de protocolos inter-niveles. Pueden existir mayores refinamientos: meta-elementos, definición de agentes específicos para realizar tareas de apoyo para el acuerdo en sí mismo (tales como *sensores*, *observadores*, *controladores*, *planificadores*, etc.), entre otros.

## 4.2 La Plataforma THOMAS

Este apartado presenta la arquitectura base para las tecnologías previamente presentadas, las que fueron concebidas para ser soportadas por un SMA abierto.

Las investigaciones actuales en la plataforma están orientadas a lograr una mayor capacidad y funcionalidad, aprovechando las características de los SMA. Por otra parte, y desde este punto de vista, los servicios son utilizados para lograr interoperabilidad, como se mencionó anteriormente. La idea principal es exportar el sistema de *agentes* como un *sistema de servicios*. El ecosistema de servicios resultante será soportado no sólo tecnológicamente sino también metodológicamente [27].

Estos conceptos se construyen sobre trabajos ya existentes: la plataforma THOMAS [3]. A continuación se resume su diseño.



**Figura 2:** Arquitectura técnica de THOMAS (inspirada en [3]).

La plataforma, incluido su middleware (Figura 2), está estructurada en tres niveles aunque no son estrictamente capas. Son ortogonalmente soportadas por cuatro componentes específicos, que están incluidos como formando parte de tres diferentes subsistemas. El subsistema *Platform Entities Management* está a su vez 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 tanto la capa de red como el componente Agent Management System (AMS). A este nivel de descripción, la plataforma es un SMA abierto.
- *Service & Organization 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.
- *Organization Execution Framework*. Es el “espacio” donde todas las entidades computacionales “viven” 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*, proporciona todas las capacidades y funciones necesarias para la gestión de agentes;
- *OMS*, hace lo propio con las organizaciones a la vez que mantiene unido al sistema en su conjunto;
- *SF*, por último, 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 pueden encontrarse en [3].

## 5 Definición del Acuerdo Emergente

Cuando se utiliza un SMA abierto para abordar un problema complejo, la solución a menudo requiere cierta adaptabilidad. Al mismo tiempo, la propia estructura debe ser flexible para lograr coordinación dentro del sistema. Y como ya se ha dicho en la introducción, la auto-adaptación es una característica cada vez más importante.

El concepto del *acuerdo entre entidades computacionales* se ve como el más adecuado para abordar la necesidad de una estructura adaptativa. El objetivo es descubrir una estructura apropiada, de modo que *emerja* como un acuerdo global. Seguidamente se definirán conceptos y elementos que, contruidos sobre estructuras básicas descriptas en los apartados anteriores, harán posible definir una organización *emergente* por sobre el *acuerdo*.

### 5.1 Guiando la Emergencia: Controles y Protocolos

En función de objetivos concretos, cualquier grupo de individuos puede ser ordenado según ciertas estructuras (por ejemplo una sociedad, una arquitectura, una jerarquía, etc.) La formación de estas estructuras se puede provocar mediante el uso de dos tipos diferentes

de elementos, los que tienen su fundamento en la limitación del rango de acciones válidas, a saber, los *controles* y *protocolos*.

Los *controles* pueden ser vistos como mecanismos que pueden tanto *imponer* como *prohibir* interacciones específicas (o conexiones arquitectónicas). Las estructuras auto-adaptativas, siendo típicamente centralizadas [2], 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 descriptos, genéricamente, como una manera de controlar las estructuras descentralizadas (o incluso distribuidas) [16]. Básicamente, cuando los protocolos están presentes cada agente conoce la forma de interactuar con el resto; es necesario acordar con ellos para poder comunicarse, pero al mismo tiempo, también *regulan* el desarrollo de la estructura de interacción.

Estos dos mecanismos definen una amplia gama de regulaciones, en los que las organizaciones de agentes y sus arquitecturas son simultáneamente reglados por controles unitarios, atómicos (como las normas, límites, bloqueos, lazos de control o restricciones), y protocolos múltiples, conectivos (concentradores, puentes, canales o espacios).

En arquitectura de software ya son conocidos muchos patrones y soluciones basadas en controles implícitos y restricciones normativas. El enfoque propuesto intenta basar la solución en el consenso: también se trata de un enfoque a nivel arquitectónico, ya que la descripción de las interacciones es pertinente 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. Por tanto, aunque no es una idea nueva sí, es muy relevante que los agentes se agrupen en organizaciones, a diferencia del clásico diseño plano de SMA, como se señaló en la Sección 3.

Una vez que una estructura primaria pueda ser definida, un grupo elemental surge como una organización preliminar, a la que se hará referencia como *una iniciativa*. Esta estructura se explica en la siguiente subsección.

## 5.2 Definición de un Acuerdo Emergente: la *Iniciativa*

Como se señaló anteriormente, se puede utilizar un conjunto de controles y protocolos para generar dinámicamente una organización preliminar dentro de un grupo de individuos (agentes, en este trabajo, aunque también podrían ser componentes genéricos).

En la 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 se llama *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 estable 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 el de THOMAS [3].

El párrafo anterior implica tres conceptos importantes:

- *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, es 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.

Este proceso puede ser visto como si el sistema se moviera a un nuevo estado, en la que la estructura del “pasado” es suplantada por una “nueva estructura emergente”. Obviamente, esta novel estructura admite nuevos elementos debido al entorno dinámico, pero ahora uno de sus objetivos es reforzar su naturaleza y tender a la persistencia.

Por tanto, uno de los objetivos de una *iniciativa* es crecer y, contraponiéndose a esto, el objetivo principal de una organización es mantenerse.

Considerando el ejemplo de Sección 2, en los primeros momentos de E1 se puede asumirse que varios coches policiales llegan al lugar, pero ninguno de ellos es 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 llamarse un *protocolo generativo* y cuando los individuos siguen este tipo de protocolo se definen patrones estructurales implícitos.

Una *iniciativa* puede generarse a partir de estos patrones, llámense *patrones del acuerdo*, y 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 para el refinamiento semántico correspondiente. Algunos ya han sido identificados (ver Tabla 1), pero es importante notar que si bien algunos de estos nombres son típicos de patrones, *no* son los clásicos patrones orientados a objeto, ya que se definen en un contexto completamente diferente: son patrones de arquitectura.

**Tabla 1.** Patrones del Acuerdo.

Nombre	Descripción y funcionamiento
Façade	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 fachada que redirige cualquier comunicación recibida; no es necesario que sea también un supervisor.
Mediator	Algún agente debe actuar como mediador mientras emerge la organización y como aún no está constituida es probable que los servicios de datos no funcionen. Posibilita el acceso a fuentes de datos, aunque sea indirectamente, y realizar las traducciones necesarias – incluyéndose las semánticas.
Surveyor	Al menos un agente debe controlar el crecimiento de la propia <i>iniciativa</i> durante el proceso de surgimiento. Puede decidir el momento para introducir nuevos elementos y también cuándo el grupo se estabilizará. El <i>surveyor</i> tiene acceso a la biblioteca de patrones – decide cuándo cierto patrón es el acertado y debe ser activado.

Hay más patrones y no todos ellos describen roles. Por ejemplo, el *Surveyor Election* define el protocolo (uno entre muchos) para decidir cuál será el próximo *surveyor*; y el *Surveyor Change* describe un protocolo para degradar al actual y transmitir su conocimiento a un nuevo *surveyor*.

Los patrones representan una estructura estática (un fragmento) que conduce a una dinámica, la *iniciativa*, alcanzando una forma estable, la organización. Como ya se señaló, el sistema es en última instancia concebido como una arquitectura *orientada a servicios*, por tanto, metodológicamente las organizaciones estables deben ser concebidas como proveedoras de ciertos servicios de alto nivel. Es así que estos servicios deben ser propuestos como punto de partida para la definición funcional de las organizaciones estáticas.

La descomposición funcional de estos servicios (o una descomposición jerárquica, desde otro punto de vista) también serán utilizados para diseñar la estructura jerárquica de las organizaciones. El concepto de *proceso de servicio*, en este contexto, se propone ofrecer una clara perspectiva semántica de la funcionalidad del servicio al describirla como un flujo de trabajo. Cada servicio (de alto nivel) se desdobra en un proceso (semántico) que describe la coordinación entre los servicios de bajo nivel; cada servicio está provisto por una organización de bajo nivel, proporcionando la descomposición estructural de la organización de alto nivel anterior. Este proceso guía la definición (semántica) de las organizaciones orientadas a servicio, y se utiliza para definir las estructuras estáticas [27].

Por otra parte, los acuerdos emergentes y dinámicos, *deben* ser coherentes con estas definiciones semánticas, de modo que este proceso proporcione también un método para descubrir los patrones necesarios para las *iniciativas*. Por supuesto, este trabajo aún tiene que ser refinado y la metodología deseada todavía tiene que adquirir una forma definitiva.

## 6 Caso de Estudio: el Ejemplo Motivador en THOMAS

A fin de aclarar el ejemplo motivador y mostrar su utilidad, en esta sección se describirán algunos de sus escenarios y algunas de las secuencias de eventos correspondientes. Estos escenarios presentan agentes que entran en el sistema (es decir, en la arquitectura adaptativa) y que juegan diferentes tipos de roles, interactuando dentro de una *unidad organizativa*. Además de proporcionar funcionalidad específica del dominio, estos escenarios también muestran comportamiento genérico a nivel de sistema que en el enfoque propuesto son las bases de la conducta adaptativa.

Como se explica en la Sección 2, con el fin de atender la emergencia E1, SUMMA112, Bomberos y Policía envían sus propios agentes.

A continuación se presentan escenarios parciales de estos agentes, utilizando una variedad de servicios integrados en THOMAS [3]. El protocolo los presenta en un orden básico:

- **Registro de una Unidad:** se utiliza para registrar una nueva unidad (unidad de organización) con una estructura específica, objetivo y unidad padres en el componente OMS (véase la Subsección 4.2)  
*RegisterUnit (UnitID, Type, Goal [ParentUnitID]).*

En el ejemplo, O1 debe ser registrada como una organización:

*RegisterUnit (O1, default, default, default)*

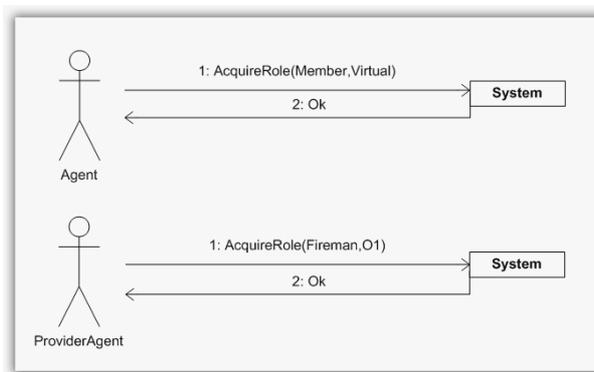
- **Registro como Miembro.** Cada agente tiene que registrarse como miembro del sistema y luego unirse a la unidad organizativa correspondiente, es decir, la organización emergente, O1. Este proceso tiene dos pasos:

1. El agente debe registrarse como miembro de la propia Plataforma:

*AcquireRole(Member, Virtual)*

2. El agente se registra en la organización con un rol específico del dominio:

*AcquireRole(Fireman, O1)*



**Figure 3:** Adquisición de un rol durante la Emergencia E1, en THOMAS.

De esta manera, la organización O1 puede incorporar todos los agentes necesarios para hacer frente a la emergencia E1. O1 se inicia como una *iniciativa* hasta que todos sus miembros han alcanzado el acuerdo necesario (un consenso interno) para convertirse en una organización estable y mantener sus características. El protocolo para decidir esto es también un patrón: puede haber muchos patrones de establecimiento de acuerdos y el *surveyor* es responsable de elegir o decidir entre ellos - por lo general a partir de información del entorno.

O1 también puede perder los agentes después de haber completado su función en E1, estos agentes pueden ser añadidos a la unidad organizacional O2 para atender la emergencia E2, como ya se explicó en el ejemplo (ver Sección 2).

- **Expulsion:** este servicio se utiliza para forzar a un agente para que abandone un rol específico.

*Expel (AgentID, UnitID, RoleID)*

Utilizando todo lo anterior, considérese por ejemplo el conjunto de acciones llevadas a cabo por un agente determinado en el ejemplo E1-E2 (p. ej. Fireman #1).

AcquireRole (Member, Virtual)	-- Activado como Fireman #1
Hello (F#1, O1)	-- Llega a O1 y se presenta
AcquireRole (Fireman, O1)	-- Se une a O1 como bombero
(...)	-- ... luego de cierto trabajo...
Expel (F#1, O1, Fireman)	-- Fireman #1 abandona O1
Hello (F#1, O2)	-- Llega a O2 y se presenta
AcquireRole (Fireman, O2)	-- Se une a O2 como bombero

Considérese la posibilidad de que el agente no necesite saber si O1 es una organización ya establecida (una organización *estática*) o que aún está creciendo (una *iniciativa*). En el primer caso, está registrada como una organización dentro del OMS, en el segundo, un agente fachada se encarga de la captura de estos mensajes. El agente bombero, en ambos casos, utiliza los servicios del sistema para enviar los mensajes relevantes.

En resumen, el sistema base (la infraestructura de *Tecnologías del Acuerdo* implementada por la plataforma THOMAS) ya proporciona los elementos necesarios para construir una arquitectura adaptable en términos de servicios internos capaces de unirse/abandonar una organización y definir la estructura en función de sus acuerdos internos. Por lo tanto, para definir un acuerdo emergente en realidad solo es necesario proporcionar el sustrato para describir una *iniciativa*, es decir, una colección de patrones estructurales del acuerdo, y el conjunto de protocolos inter-niveles que los identifican y sus roles (*fachada*, *surveyor*, etc.)

La evolución de este ejemplo constituye trabajo en progreso. Utilizando los conceptos y patrones presentados en la sección anterior, un conjunto de patrones estructurales del acuerdo actualmente se almacenan y pueden accederse a ellos en la plataforma, y se están utilizando para una serie de experimentos (incluyendo este ejemplo). Estos conceptos y los constructos que describen ya han demostrado su relevancia - pero la sintaxis actual y los protocolos no pueden considerarse aún como óptimos. Sin embargo la *performance* no es la principal preocupación en este momento, sí el esencial dinamismo y la capacidad de adaptación requeridos por la arquitectura.

## 7 Conclusiones y Trabajos Futuros

Este trabajo ha explorado los conceptos estructurales como la base de un enfoque arquitectónico para proporcionar auto-adaptabilidad a sistemas software. El concepto propuesto de *iniciativa* debe ser considerada como punto de partida para proporcionar mecanismos para cambiar los patrones de composición y tipos de elementos dentro de tales sistemas.

El dinamismo requerido puede ser soportado por un acuerdo emergente - una estructura arquitectónica evolutiva, basada en la combinación de controles y protocolos predefinidos: estos son tratados en el contexto del marco orientado a servicios, basado en agentes y centrado en organizaciones, definido en *Tecnologías del Acuerdo* y provisto por la implementación en la plataforma THOMAS.

El ejemplo motivador muestra por qué es necesario tener en cuenta un entorno de adaptación, en lugar de un "clásico" sistema cerrado o de un solo diseño de sistema abierto. Para ofrecer realmente la respuesta necesaria en un caso de emergencia, SUMMA112 tiene que coordinar con los sistemas de información del Departamento de Bomberos, la Policía y todos los hospitales de la zona. La operación es local y descentralizada. Esto implica que no es posible tener una estrategia unificada pre-programada para gestionar situaciones de emergencia, ya que debería ser incorporada en varios sistemas independientes que sólo actúan juntos en determinadas circunstancias. La idea clave es crear un contexto arquitectónico en el que los agentes son *coordinados* y *reorganizados* mediante su inclusión en estructuras preliminares -los patrones del acuerdo - y luego en organizaciones estables. La arquitectura base descrita en las Secciones 3 y 4 ya incluye todos los servicios y facilidades necesarios para llevar a cabo reconfiguraciones, y con los conceptos

propuestos, se puede considerar como punto de partida para establecer las estructuras necesarias para lograr una auto-adaptabilidad real.

Tecnológicamente, el trabajo existente es compatible con FIPA [15] y también capaz de interactuar con agentes Jade [17]. Los progresos actuales se dan a nivel de servicios con su adaptación a la plataforma OSGi [22], y a nivel de agente, afectando el rendimiento. De hecho, aún cuando el enfoque propuesto parece prometedor, en el sentido de que es posible lograr la auto-adaptación, estos son los primeros pasos y todavía queda mucho trabajo por realizar.

En los trabajos futuros se desarrollarán e implementarán variantes del enfoque propuesto, con el fin de refinar todo lo necesario. Los conceptos están evolucionando, pero incluso en esta fase inicial, los fragmentos del enfoque ya han demostrado su utilidad y su poder expresivo. Los resultados actuales sugieren que la arquitectura adaptativa es posible, y esto podría satisfacer la promesa de generalizar la utilidad y extensión del enfoque de SMA, adaptándolo a nuevas y más ágiles tecnologías.

**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. Agreement Technologies (AT) Project: <http://www.agreement-technologies.org/> (2009)
2. Andersson, J.; de Lemos, R.; Malek, S.; and Weyns, D. Modeling Dimensions of Self-Adaptive Software Systems. In *Software Engineering for Self-Adaptive Systems*, vol. 5525 Lecture Notes in Computer Science, pages 27-47, Springer, 2009.
3. Argente, E., Botti, V., Carrascosa, C., Giret, A., Julian, V., and Rebollo, M.: *An Abstract Architecture for Virtual Organizations: The THOMAS Project*. Technical report, DSIC, Universidad Politécnica de Valencia (2008).
4. Argente, E., Julian, V., and Botti, V.: Multi-Agent System Development based on Organizations. *Electronic Notes in Theoretical Computer Science* 150(3):55-71 (2006).
5. Atienza, M., Schorlemmer, M.: I-SSA - Interaction-situated Semantic Alignment. *Proc Int. Conf. on Cooperative Information Systems (CoopIS 2008)* (2008).
6. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., and Orchard, D.: *Web Services Architecture*. W3C WSA Working Group, W3 Consortium (2004)
7. Casadei, M., and Viroli, M.: Applying self-organising coordination to the emergent tuple organization in distributed networks. (Brueckner, S. et. al. editors) *2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008)*.
8. Centeno, R., Fagundes, M., Billhardt, H., and Ossowski, S.: Supporting Medical Emergencies by SMA. In *Agent and Multi-Agent Systems: Technologies and Applications*. LNCS, vol. 5559:823-833. Springer (2009).
9. Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S.: *Web Services Description Language (WSDL) 1.1*. W3C Consortium. W3C Note (2001)
10. Ciancarini, P.: Coordination models and languages as software integrators. *ACM Computing Surveys*, 28(2):300-302 (1996)

11. Cuesta, C.E., Fuente, P., Barrio, M., Beato, E. Dynamic Coordination Architecture through the use of Reflection. Proc. 16<sup>th</sup> ACM Symposium on Applied Computing (SAC 2001), pages 134-140, ACM Press, March 2001.
12. DeLoach, S.: Moving multi-agent systems from research to practice. International Journal of Agent-Oriented Software Engineering - Vol. 3, N<sup>o</sup>. 4, pages 378 – 382 (2009)
13. Esteban, J., Laskey, K., McCabe, F., and Thornton, D.: Reference Architecture for Service Oriented Architecture 1.0. Organization for the Advancement of Structured Information Standards (OASIS) (2008).
14. Fiadeiro, J. L.: Designing for Software's Social Complexity. In Computer - IEEE Computer Society, pages 34-39 (2007).
15. FIPA. FIPA Abstract Architecture Specification. Technical Report SC00001L, Foundation for Intelligent Physical Agents. FIPA TC Architecture (2002).
16. Galloway, A.R. Protocol: How Control Exists after Decentralization. MIT Press, 2004.
17. JADE - Java Agent DEvelopment Framework. <http://jade.tilab.com/>
18. Kephart, J.O. and Chess, D.M. The Vision of Autonomic Computing. IEEE Computer 36(1):41-50, January 2003.
19. Kramer, J. and Magee, J. Self-Managed Systems: an Architectural Challenge. In Future of Software Engineering (FOSE @ ICSE'2007), pages 259-268, IEEE, May 2007.
20. MacKenzie, C., Laskey, K., McCabe, F., Brown, P., and Metz, R.: Reference Model for Service Oriented Architecture 1.0. Organization for the Advancement of Structured Information Standards (OASIS) (2006).
21. OMA: The Open Mashup Alliance: <http://www.openmashup.org/>
22. OSGi: formerly known as the Open Services Gateway initiative, now an obsolete name. <http://www.osgi.org/>
23. Ossowski, S.: Co-ordination in Artificial Agent Societies. LNAI 1535. Springer (1999).
24. Ossowski, S.: Coordination in Multi-Agent Systems: Towards a Technology of Agreement. LNCS 5244. Springer (2008).
25. Papadopoulos, G. A., and Arbab, F.: Coordination models and languages. In Zelkowitz, M. V., editor, The Engineering of Large Systems, volume 46 of Advances in Computers, pages 329-400. Academic Press (1998).
26. Pérez, J. S., Cuesta, C., and Ossowski, S.: El Acuerdo como Arquitectura Adaptativa para Sistemas Multiagente Abiertos. XV Congreso Argentino de Ciencias de la Computación – CACIC (2009).
27. Pérez, J. S., Cuesta, C., and Ossowski, S.: Agreement Technologies for Adaptive, Service-Oriented Multi-Agent Systems. II Workshop on Agreement Technologies WAT – XIII Conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA (2009)
28. Rosenschein, J., and Zlotkin, G.: Rules of Encounter – Designing Conventions for Automated Negotiation among Computers. MIT Press (1994).
29. Sierra, C., Debenham, J.: Information-Based Agency. Proc Intl. Joint Conference on AI (IJCAI-2007). AAAI Press, pages 1513-1518 (2007).
30. SUMMA112: [http://www.madrid.org/cs/Satellite?language=es&pagename=SUMMA112%2FPage%2FS112\\_home](http://www.madrid.org/cs/Satellite?language=es&pagename=SUMMA112%2FPage%2FS112_home) (2009).
31. Weyns, D., Helleboogh, A., and Holvoet, T.: How to get multi-agent systems accepted in industry? International Journal of Agent-Oriented Software Engineering - Vol. 3, N<sup>o</sup>. 4, pages 383 – 390 (2009).