

# El Acuerdo como Arquitectura Adaptativa para Sistemas Multiagente Abiertos

Santiago Pérez<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> Kybele Grupo de Investigación, Depto. de Lenguajes y Sistemas Informáticos II

Universidad Rey Juan Carlos,

28933 Móstoles (Madrid), España

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

carlos.cuesta@urjc.es

**Resumen.** Los sistemas multiagente son cada vez más populares para la resolución de problemas complejos. Sin embargo, la flexibilidad y adaptabilidad frecuentemente se ven comprometidos por la necesidad de alcanzar una coordinación interna. En este trabajo se plantea una solución arquitectónica, como una nueva construcción, el *acuerdo*, diseñada para ser *adaptativa*. En la primera parte del trabajo se discute el problema de la coordinación y algunas de sus soluciones tradicionales. Luego se describe una *arquitectura base* que incluye organizaciones y servicios. Seguidamente se trata el concepto del acuerdo esbozado en un conjunto de capas lógicas y como facetas entrelazadas. Luego se presenta un caso de estudio en el dominio de las emergencias médicas donde se aplica el modelo del acuerdo y se hace énfasis en algunos requisitos para que la evolución de la configuración del sistema pase de dirigida a emergente. Finalmente se indican algunas líneas de trabajo futuro.

**Palabras claves:** Sistemas multiagentes, coordinación, acuerdo, adaptación, reconfiguración emergente.

## 1 Introducción

En los sistemas software, las tecnologías inteligentes de la información están relacionadas con las capacidades de aprender, razonar y administrar el conocimiento. El agente representa uno de sus enfoques más sofisticados. Este concepto ha evolucionado desde un simple “agente inteligente” complejo, a conjuntos de múltiples agentes con complejidad distribuida, conocidos como Sistemas Multiagente (SMA).

Teóricamente, un SMA debe poder encontrar el balance perfecto entre las reglas estrictas que definen un sistema de razonamiento (un agente) y la naturaleza adaptativa de la evolución de una “sociedad” de individuos autónomos. El primero asume un mundo cerrado y una configuración estática, y el segundo provee los medios para abrir el sistema de manera flexible, definiendo una estructura dinámica capaz de hacer frente a la complejidad social.

Al menos en la práctica, muchos SMA son menos flexibles de lo que se esperan a priori. Estabilizan una estructura concreta durante su fase de entrenamiento, la que permanecerá inalterada para mantener sus capacidades de razonamiento, y por lo tanto, no pueden

cambiar y evolucionar, limitando su *adaptabilidad*. Desde este punto de vista, el enfoque SMA no cumpliría su inicial promesa de garantizar que el sistema sea efectivamente adaptable. Esto se debe a la necesidad de una *coordinación* interna, como cada agente tiene su propia agenda, solo una acción coordinada puede asegurar un comportamiento consistente. La idea principal de este trabajo es que esta coordinación no sea provista por los propios agentes, sino por la estructura que los contiene (“arquitectura”).

En esta dirección hay varias propuestas de SMA que proveen un paso intermedio en la forma de *organizaciones virtuales*, es decir, grupos de agentes que comparten características o restricciones comunes. De esta manera, el SMA en sí puede ser dividido en fracciones más pequeñas, con agendas diferentes. La idea principal es que los agentes en una organización son coordinados automáticamente, y por tanto, se eleva el nivel de abstracción.

Obviamente, esta segunda estructura simplifica la adaptación: mayormente se reduce a la *reorganización* del agente, es decir que un agente sea segregado de una organización y se reagrupe en otra con diferentes restricciones. Esto puede traer consecuencias tanto para el agente como para las organizaciones origen y destino.

Al mismo tiempo, esta situación implica un número adicional de cuestiones. Primero, acerca del rol interno de las organizaciones en el SMA y su significancia funcional; y segundo, acerca de la necesidad de proveer coordinación (de orden superior) para las propias organizaciones, de forma que puedan conseguir su adaptación. Para responderlas, se necesitan definir dos conceptos adicionales, respectivamente, *servicios* de una organización y *acuerdos* entre ellas.

La primera provee tanto una base metodológica para el planteo, como una conexión directa a SOA [16] [25]; esto solo se mencionará brevemente.

La investigación en este trabajo se puede resumir, globalmente, como la búsqueda de tres objetivos principales, a saber:

- Transformar el clásico enfoque orientado a agentes, originalmente de un diseño cerrado de un SMA, a un ecosistema *abierto* orientado a servicios,
- Definir la infraestructura y metodología correspondientes para conseguirlo, utilizando la noción de *organizaciones* como el nexo conceptual, y
- Proveer *coordinación* interna con la definición de *acuerdo*, concebido como una construcción adaptativa a nivel arquitectónico, lo que proveería coordinación como una propiedad emergente.

El propósito final es satisfacer la promesa original de los SMA, es decir definir una tecnología *inteligente* genérica de resolución de problemas, con capacidad de ser utilizada en un contexto *abierto*, y capaz de *adaptarse* a evoluciones futuras, apoyada en una arquitectura auto-organizada.

El resto del trabajo está estructurado como sigue: primero se expone el problema de la coordinación dentro de un SMA, se discuten enfoques históricos para resolverlo, apuntando a propuestas basadas en organizaciones. Luego se presenta el Modelo del Acuerdo como solución potencial en este contexto, posteriormente, se describe la arquitectura base que incluye organizaciones. Se explican luego las cinco principales perspectivas a considerar en el modelo, así como algunas características implícitas en sus relaciones. Finalmente se provee de un ejemplo del mundo real para ilustrar la naturaleza arquitectónica y emergente del concepto, en éste y en futuros desarrollos; y se discuten brevemente los futuros pasos en esta dirección.

## 2 El Problema de la Coordinación

Como ya se ha dicho, los agentes originalmente fueron concebidos como actores solitarios, conciliando una serie de requisitos conflictivos en su interior. Con el advenimiento de los SMA se logró un cambio de enfoque y pudieron ser posibles nuevos métodos. Sin embargo el problema ahora es de *coordinación*.

Una de las definiciones de la palabra “coordinar” de la Real Academia Española [29] es “concertar medios, esfuerzos, etc., para una acción común”. En el campo de los SMA tal vez una de las definiciones más aceptadas de “coordinación” provenga de la Teoría de la Organización: “gestión de las dependencias” entre actividades organizativas [26]. Desde un punto de vista “micro” (centrado en el agente) [35] la coordinación es entendida como una *adaptación al entorno*. Si nuevos agentes entran en el entorno de otro agente, éste tiene que coordinar con aquellos, reconsiderando sus metas, planes y acciones, teniendo en cuenta las nuevas y potenciales interdependencias. Por otro lado, con una visión “macro” (centrada en el SMA), las consecuencias de la coordinación pueden ser concebidas como algo global (plan, decisión o acuerdo). Esto puede ser un plan “compartido” [30] si los agentes consiguen un acuerdo explícito durante el proceso de coordinación; o puede ser una suma de planes individuales, en ocasiones llamado “multi-plan” [27]. En este nivel, los resultados de la coordinación pueden ser evaluados como una conjunción de las metas de los agentes o teniendo en cuenta la funcionalidad del SMA como un todo.

En resumen, cuando se utiliza un SMA como una solución de software, el problema de la coordinación siempre está presente. De hecho, si se tiene una estructura de agentes auto-organizada, se la puede considerar como *óptima*, porque podría resolver los temas de coordinación. En este contexto, la hipótesis es que los agentes están adecuadamente organizados, es decir que “se coordinan solos”, como individuos en una sociedad (humana). Esta auto-organización, podría ser, al menos en parte, emergente. La idea es que, deberían organizarse de manera que “espontáneamente” reaccionen como se desea, “moviéndose” en la dirección correcta. Pero para que esto sea satisfecho, el enfoque orientado al agente debe ser complementado con una estructura de coordinación auto-organizada, esto es, a nivel de arquitectura.

### 2.1 Evolución de los Enfoques de Coordinación

Los trabajos pioneros, donde se relacionan agentes y coordinación, fueron abordados por la Inteligencia Artificial (IA) a finales de los años 70 y comienzos de los 80. En estos trabajos la *cooperación* es vista como un concepto requerido para la solución inteligente de problemas complejos [2].

En los inicios de las investigaciones y desarrollos en sistemas cooperativos, algunas de las cuestiones más importantes fueron cómo encontrar los *modelos cooperativos* para cada clase de problema, cómo *representar* estos modelos de manera que se puedan *ejecutar*, y cuáles eran los *métodos* necesarios para que los sistemas cooperativos fueran realmente efectivos.

Se desarrollaron varias soluciones para resolver el problema de la cooperación. La arquitectura *pizarra* [15] provee cooperación entre fuentes de conocimiento usando un mecanismo sencillo de comunicación. Cada subsistema toma datos de la pizarra, produce sus resultados y luego los escribe en la pizarra. La *red de contratos* [32] propone la *negociación* como mecanismo para coordinar y asignar tareas a las diferentes entidades que participan en la resolución del problema. Cada entidad puede ser *contratante* o *contratada*,

y las condiciones de cooperación se establecen al firmar el contrato. Otra solución fue la *arquitectura reactiva*. En ésta se trata de obtener un comportamiento inteligente desde modelos simples, sin mecanismos de representación de conocimiento, razonamiento o aprendizaje [8]. El énfasis en estos modelos está en las *interacciones*, de las cuales *emerge* el comportamiento grupal para resolver los problemas. Finalmente aparecen las *arquitecturas de agentes con capacidad organizacional*. Aparte del conocimiento del dominio, los agentes necesitan conocer sus propias características *sociales*. Este conocimiento es importante porque les permite saber cuál es su rol en la organización, las reglas y las capacidades de los otros agentes.

## 2.2 SMA Orientados al Agente vs. SMA Orientados a la Organización

Los SMA pueden ser diseñados con una estructura organizacional fija, los que estarán compuestos por agentes *estáticos*. Utilizando agentes flexibles o evolutivos se pueden obtener modelos organizacionales que simulan organizaciones humanas. Para lograr esto es necesario que se puedan modificar algunas propiedades (p. ej. los roles, niveles de autonomía o mecanismos de control).

Las metodologías de SMA puramente *orientadas al agente* (como MAS-CommonKADS [23], Gaia [37], MaSE [36], Tropos [22] o Prometheus [28], entre otras) se concentran usualmente en la visión del agente; se asume que el comportamiento final del sistema *emerge* de las interrelaciones entre los agentes diseñados. El comportamiento global no se analiza en detalle. Si agentes externos ingresan a la organización, el comportamiento global del sistema puede verse afectado a menos que se establezcan controles, normas, sanciones, etc.

Por otro lado, las metodologías de SMA *orientadas a las organizaciones*, tienen en cuenta la organización del sistema desde los inicios. El análisis se hace desde una perspectiva global (Agent-Group-Role [18], MESSAGE [9], ANEMONA [21], AML [11], OperA [14], Civil Agent Societies [13], MOISE [20], Electronic Institutions [17], HARMONIA [34], GORMAS [3], entre otras). Los objetivos describen los propósitos organizacionales a un nivel alto. Esto permite la determinación de tareas, tipos de agentes, asignación de recursos entre miembros, etc. El sistema se estructura por los roles, interacción entre agentes y los lenguajes de comunicación que utilizan. Las normas sociales describen el comportamiento deseado de los miembros. Estas normas luego derivan en el control, prohibiciones, sanciones, etc. para lograr el comportamiento global esperado. Se tienen en cuenta mecanismos para permitir que agentes externos entren en la organización y controlar su comportamiento. Esta característica es particularmente útil para diseñar SMA *abiertos*.

## 2.3 Hacia el Modelo de Acuerdo

Genéricamente, los individuos se organizan en estructuras de alguna clase, tanto en una sociedad como en una arquitectura, utilizando dos clases diferentes de esquemas. Estos son, por un lado, los *controles*, que *hacen cumplir* o *prohíben* interacciones específicas; y por otro, los *protocolos*, que las *permiten* o *dirigen*. Los primeros están basados en la fuerza o la *imposición* y los últimos en el consenso y el *acuerdo*.

El concepto de *acuerdo* entre entidades computacionales parece ser un enfoque adecuado para enfrentar este problema. El objetivo es “descubrir” una estructura conveniente de controles y protocolos de forma que *emerja* como una estructura global, *el*

*acuerdo*. Por consiguiente, un concepto importante, aunque no sea tan nuevo, es que los agentes están agrupados en *organizaciones*, a diferencia del clásico esquema de los SMA. A medida que las estructuras de agentes se van volviendo cada vez más complejas está claro que, para cierta clase de problemas, no se necesita una súper estructura, como la pizarra. Se necesita que los agentes se *organicen por sí mismos* en organizaciones y luego de esto, en un nivel superior, en organizaciones basadas en acuerdos. El objetivo principal es evolucionar de esta coordinación emergente a un *acuerdo emergente* entre las entidades involucradas en una solución.

Es importante hacer notar que el concepto de *acuerdo* está en desarrollo, y aunque hay elementos que lo definen (ver Sección 4), el concepto no está cerrado. El acuerdo está en constante evolución, así como la estructura que lo define [1].

### 3 Una Arquitectura Base para el Modelo de Acuerdo

La arquitectura que dé apoyo al modelo se define como un *SMA abierto* y también como *orientada a servicios (AOS)*, centrada en organizaciones y basada en agentes. La idea central es la de *servicio*, el componente básico de la arquitectura es el *agente*, y la estructura que los une es la *organización*. Esta naturaleza multinivel y de múltiples puntos de vista debe ser específicamente habilitada por la arquitectura técnica. El lugar lógico para proveer este soporte es el “middleware”, pues la plataforma se concibe como un sistema distribuido.

Para que se cumplan ambos conceptos al mismo tiempo, que sea un SMA y una AOS, una de estas estructuras y conceptos centrales (agente, servicio) debe ser la base de la otra. La alternativa elegida es la de agentes soportando servicios. Primero, los agentes son entidades computacionales bien conocidas, implican cierta granularidad y cumplen con estándares existentes [19]. Por otro lado, los servicios aún están definiendo su rol y relevancia, aunque ya tienen numerosos estándares [7, 12, 16, 25], éstos están definidos en diferentes niveles de granularidad y la mayoría solo imponen restricciones en las interfaces de los servicios. Por tanto, es fácil concebir un servicio como una manera de presentar las capacidades operacionales de un agente, incluso de un grupo de agentes, jerárquicamente amalgamados en una organización. Resumiendo, la elección obvia es tener una plataforma definida como una AOS, construida por sobre un SMA.

Estos conceptos se construyen sobre un trabajo existente, como lo es THOMAS [4] y GORMAS [3]. En la Figura 1, la arquitectura THOMAS, incluyendo su middleware, está estructurada en tres niveles aunque no son exactamente capas. Son ortogonalmente soportadas por cuatro componentes, organizados en diferentes subsistemas y proveen las capacidades a los diferentes niveles. Estos son:

- *Platform Kernel*. El nivel inferior que soporta todo lo de arriba, es el kernel del middleware. A este nivel de descripción, la plataforma es un SMA abierto.
- *Service & Organization Management*. Nivel conceptual compuesto por el OMS y el SF. No es estrictamente un subsistema.
- *Organization Execution Framework*. Es el “espacio” donde “viven” las entidades computacionales y donde realizan sus tareas.

Los componentes principales, que proveen al middleware con la mayoría de sus capacidades son:

- *Agent Management System (AMS)*. Provee las capacidades y funcionalidades requeridas por los agentes.

- *Organization Management System (OMS)*. Provee las capacidades y funcionalidades requeridas por las organizaciones.
- *Service Facilitator (SF)*. Provee las capacidades y funcionalidades que se requieren para que una selección de operaciones en una organización se comporte como un servicio unificado.

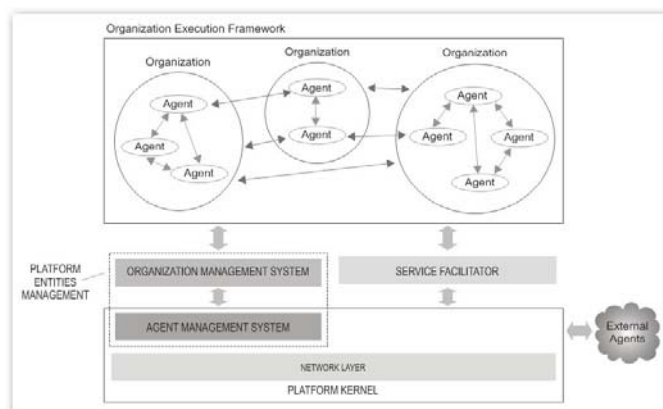


Figura 1: THOMAS Arquitectura Técnica (inspirada en [4])

Esta arquitectura provee una explicación satisfactoria de la perspectiva multi-nivel y multi-aspecto que permite conciliar las dos visiones previamente descritas. Esta plataforma se presenta como un punto de partida para el desarrollo propuesto de una estructura de *coordinación basada en acuerdos*, y es suficiente para simular y probar variantes del modelo de acuerdo.

### 3.1 El Rol de los Servicios

El concepto de servicio es la piedra angular de la arquitectura. Un servicio tiene siempre una naturaleza relacionada con el comportamiento y presenta una interfaz funcional. Un servicio es brindado por un proveedor de servicios y por un rol concreto de este proveedor.

Básicamente hay dos tipos de servicios:

- *Servicios Base* o simplemente *Servicios*. Son a nivel usuario, definidos para una aplicación concreta. El propósito de la plataforma es servir de soporte para definir y usar servicios de esta clase.
- *Servicios de Sistema*. No son estrictamente “servicios” ya que no son ofrecidos por un proveedor concreto a nivel de usuario, y están usualmente a un nivel más bajo que los anteriores.

De acuerdo a su función y a la extensión de sus capacidades se pueden identificar tres conjuntos de servicios:

- *Servicios Estructurales*: definen una estructura organizacional/arquitectónica.
- *Servicios de Información*: proveen información específica acerca de componentes en una organización.
- *Servicios Dinámicos*: permiten, por ejemplo, que las entidades entren o abandonen dinámicamente una organización.

Puede haber varias implementaciones para el mismo servicio, descritas usando la misma interfaz y el mismo perfil semántico. Al mismo tiempo cada implementación es

semánticamente especificada por un modelo de proceso de servicio diferente. A su vez, cada implementación, conforme al modelo de proceso, puede ser provista por diferentes proveedores de servicios. El proceso de servicio es también la base para la definición de servicios compuestos. Aunque aquí el enfoque tiene una naturaleza semántica, es esencialmente el mismo que se utiliza, desde una perspectiva de comportamiento, en el contexto de la composición de servicios basada en orquestación [24].

### 3.2 El Rol de las Organizaciones

El elemento activo más importante y el concepto unificador de la arquitectura es la *organización*. La jerarquía organizacional es la que hace posible definir, simultáneamente, la arquitectura como orientada a servicios (a alto nivel) y basada en agentes (a bajo nivel). La noción de organización define una estructura recursiva. Una organización está compuesta por *unidades* o *unidades organizacionales*. Una unidad es una entidad activa con un comportamiento definido y externamente observable, y tiene una naturaleza colectiva (donde la unidad es una organización en sí misma) o una autónoma (cuando la unidad es también un agente). Cuando la unidad es también una organización, se ve la estructura recursiva.

De hecho, el modelo se describe como basado en organizaciones y la unidad provee soporte para el resto de ellas. En esta concepción, la unidad es el substrato que apoya la reunión de agentes y la definición de servicios.

## 4 El Acuerdo como una Estructura de Capas y Multi-facetada

La noción central en el enfoque propuesto es el *acuerdo* entre entidades computacionales, concebido como una construcción arquitectónica. Deben considerarse temas claves para proponer la *coordinación basada en acuerdo*. Estos pueden ser vistos como una estructura en “torre” o “pirámide” [1] donde cada nivel provee funcionalidad y entradas al nivel que se encuentra por encima. Por consiguiente, el acuerdo debe ser visto como una estructura de *capas*, por definición. Intuitivamente puede verse que si una organización llega a un acuerdo, los agentes agrupados en ella deben respetar los términos. Las capas, de abajo hacia arriba, son:

**Semántica.** Es la capa inferior, ya que las cuestiones semánticas influyen sobre todas las otras. La alineación semántica de ontologías [6] es muy importante para evitar las no correspondencias y es necesaria para tener un entendimiento común.

**Normas.** La siguiente capa tiene que ver con la definición de reglas que determinan restricciones que deben satisfacer los acuerdos y los procesos para llegar a ellos. Pueden implicar roles estructurales que afectan (o controlan) el comportamiento de los agentes.

**Organizaciones.** Implican una súper-estructura que restringe la manera en que se consiguen los acuerdos, fijando la estructura social de los agentes, las capacidades de sus roles y las relaciones entre ellos [5].

**Argumentación y Negociación.** Pueden ser vistos como protocolos que definen la estructura de un acuerdo.

**Confianza.** Es la capa superior de la torre. Los agentes necesitan mecanismos de confianza que resuman la historia de acuerdos y las ejecuciones subsecuentes de los acuerdos para construir relaciones a largo plazo entre ellos [31]. Un acuerdo durable crea la sensación de confianza y da una noción de reputación.

Puede concebirse un cambio de visión a una figura multi-facetada, ya que el *acuerdo* impregna todas las facetas/niveles. En este sentido, las facetas se entrelazan (Figura 2), pero el acuerdo aún continúa siendo una estructura de capas.

El acuerdo es como una estructura transversal, que mantiene una relación bidireccional con los elementos que contiene.

Es importante notar que la perspectiva multi-facetada (el “pentágono”) no intenta reemplazar a la torre. La arquitectura descrita previamente aún es jerárquica en muchos sentidos, pero el *acuerdo* en sí mismo puede considerarse no solo en capas sino también como multi-facetado. Las capas se conciben para proveer separación lógica de conceptos.

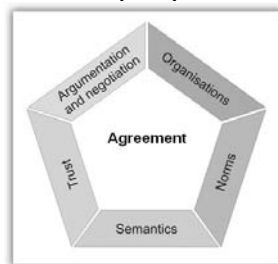


Figura 3: Estructura multi-facetada de un Acuerdo

## 5 Caso de Estudio: el Sistema *mHealth*

En esta sección se presenta un caso de estudio para ilustrar la naturaleza auto-adaptativa de la construcción del acuerdo y los pasos de la metodología en desarrollo. Este ejemplo está relacionado con el *demostrador mHealth* (mobile-Health), un prototipo evolutivo que está en desarrollo en el proyecto Agreement Technologies [1] en el dominio de las emergencias médicas. Las emergencias médicas en la Región Autónoma de Madrid son gestionadas por la organización SUMMA112 [33]. Se describe una emergencia (E1) en el sistema y el escenario tiene que evolucionar para reaccionar ante una segunda (E2). Esto describe cómo la adaptabilidad requiere una *reorganización de agentes* dentro del *acuerdo*.

**E1:** un incendio en la “Casa de Campo” (parque urbano al oeste de Madrid). De las 500 personas allí presentes 65 presentan síntomas de asfixia. SUMMA112 recibe la información e inicia su trabajo. Luego de analizar la información decide que 5 ambulancias y un helicóptero son necesarios. La coordinación con Bomberos y Policía también es urgente en esta situación. Estas entidades enviarán 3 camiones bomberos y 5 coches policiales. Desde un enfoque *organizacional* estos elementos forman la organización, **O1**. Considerando el escenario como un entorno SMA, cada actor sería un agente. Así, tenemos 14 agentes interactuando en la organización O1.

**E2:** Una hora más tarde se produce un accidente de coches en cadena (E2) en una carretera próxima al lugar de E1. 7 coches se han accidentado y 2 de ellos están incendiados. SUMMA112 decide que son necesarias 3 ambulancias y que deben contactar con los hospitales cercanos. Luego de coordinar con Bomberos y Policía, estas entidades deciden enviar un camión bombero y 3 coches policiales. Todos estos elementos (inicialmente 7), forman una segunda organización, **O2**.

Para tener en cuenta potenciales interacciones entre ambas emergencias debemos considerar ambas organizaciones. En O1 todos los elementos interactúan de un modo coordinado para enfrentar la emergencia. Estos elementos llegan a un *acuerdo* para ello.



Así, la construcción del *acuerdo* puede ser definida como “el conjunto de elementos que interactúan de una manera coordinada para resolver un problema”. Para asignar recursos a E2 se tiene en cuenta que algunos recursos que previamente fueron asignados a O1 ahora pueden pasar a O2 ya que las condiciones en E1 pueden haber cambiado durante la última hora. Este proceso implica una *reconfiguración* de O1. Metodológicamente, algunos servicios que fueran provistos para O1 ya no son necesarios para E1, y ahora son provistos sobre O2. Por ejemplo, el sistema decide que uno de los camiones bomberos ya no es necesario en E1 y puede ser asignado a E2. La O1, ahora con menos elementos, continúa trabajando sobre E1; y un nuevo acuerdo es creado alrededor de E2, definiendo O2. Al mismo tiempo, se crea un *acuerdo* mayor que abarca ambas unidades (y por lo tanto, define otra unidad). Este acuerdo continuaría adaptándose a los cambios.

Una reconfiguración gestionada manualmente es una *evolución dirigida*. Idealmente y en un entorno SMA, la meta es lograr una *reconfiguración automática*. El ecosistema global debe realizar una serie de reorganizaciones hasta encontrar un punto óptimo y detenerse. Esto puede ser un proceso continuo, ya que la situación misma puede evolucionar.

El criterio utilizado para decidir si un agente pertenece a un acuerdo debe ser gestionado de la misma manera: esto define un *acuerdo emergente*, donde la estructura misma emerge de esta situación.

El caso de SUMMA112 muestra claramente porqué se necesita considerar un *ecosistema* general. Para proveer la respuesta requerida en una emergencia se tienen que coordinar los sistemas de información de las distintas entidades. Esto implica que no es posible tener una estrategia unificada *pre-programada* para gestionar emergencias en la Región de Madrid.

En resumen, el ejemplo describe cómo lograr coordinación dentro de un *ecosistema* de servicios (o un sistema de sistemas), y también justifica porqué este comportamiento no puede ser completamente pre-diseñado, y por lo tanto debe ser *emergente*.

## 6 Conclusiones y Trabajos Futuros

En este trabajo se ha descrito el problema implícito de la coordinación que compromete la adaptabilidad en el enfoque de los SMA, y se ha propuesto una construcción a nivel arquitectónico como solución a este problema: el *acuerdo*.

El ejemplo describe sucintamente un caso actual (simulado) de esfuerzo de coordinación en el sistema real SUMMA112. Existe un SMA estructurado en organizaciones e implementado sobre la arquitectura descrita en [10].

El proceso de reconfiguración ha sido también modelado y probado usando diferentes enfoques, pero este proceso manual es solo el primer paso en la investigación. El siguiente es desarrollar un enfoque *model-driven* para guiar la reconfiguración, y continuará con un enfoque de auto-adaptación emergente, lo que constituye el objetivo final.

La idea clave en el Modelo de Acuerdo es que crea un *contexto* arquitectónico, en el que los agentes (y organizaciones, e incluso los servicios que proveen) son coordinados y reorganizados. En particular no hay un elemento arquitectónico encargado de la reconfiguración, en su lugar, cada propiedad se concibe como *emergente*.

Los futuros trabajos desarrollarán e implementarán variantes del modelo del acuerdo. Los resultados actuales sugieren que la arquitectura adaptativa es ciertamente factible y podría satisfacer la promesa de generalizar la utilidad y extensión del enfoque SMA.

## Reconocimientos

Este trabajo ha sido parcialmente financiado por el Proyecto AT (CONSOLIDER CSD2007-0022, INGENIO 2010) del Ministerio de Ciencia e Innovación de España, y por el COST Action AT (COST IC0801) del EU RTD Framework Programme.

## Referencias

1. Agreement Technologies (AT) Project: <http://www.agreement-technologies.org/> (2009)
2. Ana Mas: *Agentes Software and Sistemas Multi-Agente: Conceptos, Arquitecturas y Aplicaciones*. Prentice-Hall (2005).
3. Argente, E.: *GORMAS: Guidelines for ORganization-based Multiagent Systems*. PhD thesis, Universidad Politécnica de Valencia (2008).
4. 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).
5. 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).
6. Atienza, M., Schorlemmer, M.: *I-SSA - Interaction-situated Semantic Alignment*. *Proc Int. Conf. on Cooperative Information Systems (CoopIS 2008)* (2008).
7. 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)
8. Brooks, R.: *Intelligence without Representation*. *Art. Intelligence*, 47:139-159 (1991).
9. Caire, G., Coulier, W., Garijo, F., Gomez, J., Pavon, J., Leal, F., Chainho, P., Kearney, P., Stark, J., Evans, R., and Massonet, P.: *Agent-oriented analysis using MESSAGE /UML*. LNCS vol. 2222:119–125 (2002).
10. Centeno, R., Fagundes, M., Billhardt, H., and Ossowski, S.: *Supporting Medical Emergencies by MAS*. In “*Agent and Multi-Agent Systems: Technologies and Applications*”. LNCS, vol. 5559:823-833. Springer (2009).
11. Cervenka, R., and Trencansky, I.: *AML. The Agent Modeling Language*. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkauer (2007).
12. Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S.: *Web Services Description Language (WSDL) 1.1*. W3C Consortium. W3C Note (2001)
13. Dellarocas, C., and Klein, M.: *Civil agent societies: Tools for inventing open agent-mediated electronic marketplaces*. In *ACM Conf. Electronic Commerce (EC-99)* (1999).
14. Dignum, V.: *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Utrecht University.
15. Erman, L., Hayes-Roth, F., Lesser, V., Reddy, R.: *The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty*. *ACM Computing Surveys* 12(2), pages 213-253 (1980)
16. 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).
17. Esteva, M., Rodriguez, J., Sierra, C., Garcia, P., and Arcos, J.: *On the Formal Specification of Electronic Institutions*. *Agent Mediated Electronic Commerce 1991*, pages 126–147 (2001)
18. Ferber, J., Gutknecht, O., and Michel, F.: *From Agents to Organizations: an Organizational View of Multi-Agent Systems*. In *Proc. AAMAS03 - Agent-Oriented Software Engineering Workshop (AOSE)* (2003).
19. FIPA. *FIPA Abstract Architecture Specification*. Technical Report SC00001L, Foundation for Intelligent Physical Agents. FIPA TC Architecture (2002).

20. Gateau, B., Boissier, O., Khadraoui, D., and Dubois, E.: MOISE-Inst: An Organizational model for specifying rights and duties of autonomous agents. In der Torre, L. V., and Boella, G., eds., First Intl. Workshop on Coordination and Organisation (2005).
21. Giret, A.: ANEMONA: Una metodología multi-agente para sistemas holónicos de fabricación. PhD thesis, Universidad Politécnica de Valencia (2005).
22. Giunchiglia, F., Mylopoulos, J., and Perini, A.: The Tropos Software Development Methodology: Processes, Models and Diagrams. In Proc. Workshop on Agent Oriented Software Engineering (AOSE), 63–74 (2002).
23. Iglesias, A., Garijo, M., Gonzalez, J., and Velasco, J.: A methodological proposal for multiagent systems development extending CommonKADS. In Proc. 10th Banff Workshop Knowledge Acquisition for Knowledge-Based Systems (1996).
24. Jordan, D., Evdemon, J., Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guizar, A., Kartha, N., Kevin Liu, C., Khalaf, R., Koenig, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P., and Yiu, A.: Web Services Business Process Execution Language (WSBP EL) 2.0. Organization for the Advancement of Structured Information Standards (OASIS) (2007).
25. 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).
26. Malone, T., Crowston, K.: The Interdisciplinary Study of Co-ordination. Computing Surveys 26 (1). ACM Press, pages 87–119 (1994).
27. Ossowski, S.: Co-ordination in Artificial Agent Societies, LNAI 1535. Springer (1999).
28. Padgham, L., and Winikoff, M.: Prometheus: A Methodology for Developing Intelligent Agents. In Proc. Agent Oriented Software Engineering (AOSE), 135–145 (2002).
29. Real Academia Española: [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=coordinar](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=coordinar) (2009)
30. Rosenschein, J., and Zlotkin, G.: Rules of Encounter – Designing Conventions for Automated Negotiation among Computers. MIT Press (1994).
31. Sierra, C., Debenham, J.: Information-Based Agency. Proc Intl. Joint Conference on AI (IJCAI-2007). AAAI Press, pages 1513-1518 (2007).
32. Smith, R.: A Framework for Problem Solving in a Distributed Processing Environment. PhD thesis, Stanford University (1978).
33. SUMMA112: [http://www.madrid.org/cs/Satellite?language=es&pagename=SUMMA112%2FPPage%2FS112\\_home](http://www.madrid.org/cs/Satellite?language=es&pagename=SUMMA112%2FPPage%2FS112_home) (2009).
34. Vazquez-Salceda, J., and Dignum, F.: Modelling Electronic Organizations. Lecture Notes in Artificial Intelligence 2691:584–593 (2003).
35. Von Martial, F.: Co-ordinating Plans of Autonomous Agents. LNAI 610, Springer (1992)
36. Wood, M., DeLoach, S., and Sparkman, C.: Multiagent system engineering. Journal of Software Engineering and Knowledge Engineering 11:231–258 (2001).
37. Wooldridge, M., Jennings, N., and Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. J. Autonomous Agent and Multi-Agent Systems 3:285–312 (2000).