

UNIVERSIDAD POLITÉCNICA DE VALENCIA  
Depto. Sistemas Informáticos y Computación

TESIS DE MÁSTER

Modelado y diseño de Sistemas  
Multiagente centrados en  
organizaciones

Autor: Sergio Esparcia García  
Dirigida por: Dra. Estefanía Argente Villaplana

Grupo de Tecnología Informática - Inteligencia Artificial  
*Departamento de Sistemas Informáticos y Computación*  
*Universidad Politécnica de Valencia*  
*Camino de Vera, s/n*  
*46022 Valencia, Spain*  
16 de Junio de 2010





# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación y objetivos . . . . .	2
<b>2. OCMAS</b>	<b>5</b>
2.1. Dimensiones de la organización . . . . .	6
2.2. Lenguajes de Modelado de Organizaciones . . . . .	11
2.2.1. Virtual Organization Model . . . . .	18
2.3. Formalización . . . . .	20
2.4. Artefactos . . . . .	27
2.4.1. Taxonomía funcional de artefactos . . . . .	29
2.4.2. Relaciones entre categorías . . . . .	33
2.4.3. Discusión . . . . .	35
2.5. Mecanismos organizativos . . . . .	36
2.6. Adaptación de OCMAS . . . . .	38
2.6.1. Concepto de adaptación . . . . .	38
2.6.2. Self-organization . . . . .	42
2.6.3. Reorganización . . . . .	46
2.6.4. Discusión . . . . .	50
<b>3. Mecanismos organizativos como artefactos</b>	<b>53</b>
3.1. Conceptos básicos . . . . .	54
3.2. Artefactos de mecanismos organizativos . . . . .	56
3.2.1. Artefactos informativos . . . . .	56
3.2.2. Artefactos incentivos . . . . .	59
3.2.3. Artefactos coactivos . . . . .	61
3.3. Trabajo relacionado . . . . .	63
3.4. Discusión . . . . .	67
<b>4. Extensión del Virtual Organization Model</b>	<b>69</b>
4.1. Dimensión Estructural . . . . .	70
4.2. Dimensión Funcional . . . . .	73

4.3.	Dimensión Dinámica . . . . .	76
4.4.	Dimensión del Entorno . . . . .	77
4.5.	Dimensión Normativa . . . . .	80
4.6.	Discusión . . . . .	81
<b>5.</b>	<b>Formalización de una VO</b>	<b>83</b>
5.1.	Especificación Organizativa . . . . .	84
5.1.1.	Dimensión Estructural . . . . .	84
5.1.2.	Dimensión Funcional . . . . .	87
5.1.3.	Dimensión del Entorno . . . . .	89
5.1.4.	Dimensión Normativa . . . . .	90
5.2.	Entidad Organizativa . . . . .	92
5.3.	Dinámica Organizativa . . . . .	92
5.4.	Sistema Multiagente basado en Organizaciones Virtuales . . .	94
5.5.	Discusión . . . . .	95
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>97</b>
6.1.	Aportaciones . . . . .	97
6.2.	Trabajo futuro . . . . .	98
6.3.	Publicaciones relacionadas . . . . .	99

# Capítulo 1

## Introducción

En los últimos años, los Sistemas Multiagente (SMA) se han convertido en una de las aproximaciones de desarrollo de software que más interés han suscitado por parte de los investigadores. Un SMA se basa en el concepto de agente, una entidad software proactiva, capaz de tomar sus propias decisiones, reactiva, capaz de reaccionar a determinadas acciones que ocurran sobre él o sobre su entorno, y social, porque un agente está capacitado para comunicarse y relacionarse con otros agentes. La unión de varios agentes es lo que da lugar a los mencionados SMA.

Para desarrollar estos sistemas es necesario disponer de herramientas, métodos y procesos que permitan en un primer momento realizar un buen análisis de los requisitos del sistema, para después diseñar el sistema de forma correcta y finalmente llevar a cabo la implementación del mismo. En un primer momento, los SMA desarrollados usando estos métodos contemplaban al agente como entidad central del sistema, pero más tarde se tomó en cuenta el concepto de organización de agentes. El agente individual pasaba a un segundo plano, y eran las organizaciones las que centraban el desarrollo del sistema, teniendo que definir sus elementos propios como estructura, normas u objetivos. Estos sistemas se conocen como Sistemas Multiagente Centrados en la Organización (OCMAS) y tienen la particularidad de ser sistemas abiertos, con lo que un agente externo al sistema podría acceder a él. Un tipo particular de OCMAS son las Organizaciones Virtuales, una entidad que define normas, estructura, objetivos y otros elementos organizativos.

Numerosos proyectos han estudiado el desarrollo de este tipo de sistemas. Uno de ellos, llevado a cabo por nuestro grupo de investigación, el Grupo de Tecnología Informática - Inteligencia Artificial (GTI) del Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia, junto al Grupo de Biomedicina, Sistemas Informáticos Inteligentes y Tecnología Educativa (BISITE) de la Universidad de Salamanca y al Grupo

de Investigación en Inteligencia Artificial de la Universidad Rey Juan Carlos, fue el proyecto 'THOMAS: Métodos, Técnicas y Herramientas para Sistemas Multiagentes Abiertos' (referencia TIN2006-14630-C03). El principal objetivo de este proyecto era el de proveer tecnología basada en agentes/sistemas multiagente, necesaria para el desarrollo de organizaciones virtuales en entornos abiertos.

La presente tesis de máster se enmarcará dentro de la continuación del proyecto THOMAS, denominada 'OVAMAH: Organizaciones Virtuales Adaptativas: Mecanismos, Arquitecturas y Herramientas' (referencia TIN2009-13839-C03-01) centrado en proveer de herramientas y metodologías que permitan a las Organizaciones Virtuales modificarse ante cambios en su entorno o sus requisitos. La organización debe ser capaz de detectar situaciones que puedan serle adversas para intentar paliarlas o situaciones que puedan ser beneficiosas para tratar de obtener ventaja de las mismas.

## 1.1. Motivación y objetivos

Las herramientas y métodos de los que se dispone para desarrollar organizaciones virtuales se encuentran en constante evolución para adaptarse a los nuevos avances en esta materia. Es por ello que las nuevas propuestas surgidas para diseñar OCMAS están basadas en propuestas anteriores, extendiendo y mejorando sus posibilidades. Con la realización de esta tesis de máster se persigue estudiar métodos de diseño y modelado de OCMAS, para posteriormente exponer una nueva propuesta en este ámbito. Este trabajo será el punto de partida a una futura tesis doctoral, que estará enmarcada también en el proyecto OVAMAH.

Concretamente, los objetivos de esta tesis de máster son:

- Definir el concepto de Sistema Multiagente Centrado en la Organización, así como estudiar las dimensiones en las que se estructura una organización de agentes.
- Dar una visión de las formas de modelar un OCMAS. Por una parte se hará un estudio de los Lenguajes de Modelado Organizativo (OML), haciendo especial hincapié en el Virtual Organization Model, y por otra parte se estudiarán distintas propuestas para modelar un OCMAS de manera formal.
- Estudiar dos elementos que ayudan a mejorar la coordinación entre los agentes que pueblan un OCMAS: los artefactos y los mecanismos organizativos.

- Presentar una extensión del Virtual Organization Model que incluya los conceptos de artefacto y mecanismo organizativo, así como diversas mejoras para el metamodelo.
- Proponer un modelo formal para definir Organizaciones Virtuales.
- Presentar los conceptos y las propuestas existentes más importantes sobre adaptación y reorganización en Sistemas Multiagente.

Para alcanzar estos objetivos, el capítulo 2 dará una descripción de los OCMAS, describiendo las dimensiones propias de una organización de agentes. Este capítulo mostrará los trabajos más relevantes en modelado de OCMAS, ya sea mediante metamodelos o formalizaciones. Finalmente, este capítulo presentará los conceptos de artefacto y mecanismo organizativo, además de un estado del arte de las propuestas sobre adaptación en sistemas multiagente. El capítulo 3 presentará nuestra propuesta para modelar artefactos como mecanismos organizativos. Usando este concepto, además de diversas mejoras, el capítulo 4 presenta una extensión del Virtual Organization Model. Nuestra propuesta para un modelado formal de organizaciones virtuales se presentará en el capítulo 5. Finalmente, el capítulo 6 dará nuestras conclusiones sobre esta tesis de máster, además de trabajos futuros en esta dirección.





## Capítulo 2

# Sistemas Multiagente Centrados en Organizaciones

Las metodologías de diseño de sistemas multiagente (SMA) se centraron inicialmente en el concepto propio de agente, de forma que todo el análisis y diseño del sistema se contempla desde la perspectiva de agentes individuales que interaccionaban entre si. Por tanto, en estas metodologías, el concepto de organización de agentes [DW04] no se definía de forma explícita, sino que la estructura quedaba definida de manera implícita mediante las relaciones y necesidades internas de los agentes que pueblan el sistema. Este tipo de metodologías, conocidas como "Metodologías SMA centradas en agentes", permite diseñar sistemas multiagente cerrados, en los que los agentes del sistema poseen comportamientos y funciones totalmente definidos, siendo cooperativos y benevolentes. Sin embargo, en estos sistemas no existe la posibilidad de que agentes externos o con comportamientos que no se ajusten a las reglas puedan participar en el sistema [ZJW01].

En los últimos años, la investigación en SMA tiende a diseñar sistemas abiertos [GPL07], en los que se permiten que agentes externos entren al sistema, teniendo los mismos derechos de acceso a las funcionalidades propias del sistema. Estos agentes externos podrían tener objetivos totalmente distintos a los de los agentes internos del sistema, promulgando comportamientos interesados que no sean ni benevolentes ni cooperativos con el resto del sistema.

Para regular un SMA abierto se adopta el concepto de organización como una entidad del sistema que posee sus propios objetivos (llamados objetivos o metas globales) definidos de forma explícita y que los agentes que se integran en la organización deben ayudar a alcanzar. Para asegurarse de que esto suceda, la organización incorpora normas y restricciones que permitan regular los comportamientos de los agentes, internos o externos, que pueblan el sistema. El cumplimiento o violación de las normas provocará que el agen-

te sea recompensado o castigado, según corresponda. Además, el sistema se encuentra organizado siguiendo una estructura o topología que los agentes adoptarán y que será útil para definir las líneas de comunicación y de jerarquía dentro del SMA. Por tanto, las metodologías de SMA recientes se focalizan en el diseño de sistemas multiagente centrados en organizaciones (Organization Centered Multi-Agent Systems, OCMAS).

Las organizaciones de agentes se pueden analizar desde diferentes puntos de vista, llamados dimensiones. Cada una de ellas considerará distintos aspectos del SMA. Estas dimensiones se describen en la siguiente subsección.

Este capítulo mostrará una visión general de los sistemas multiagente centrados en organizaciones, presentando algunos de los trabajos más importantes en el diseño y modelado de este tipo de SMA, que han servido como base para las propuestas realizadas en esta tesis de máster. Para comenzar, la sección 2.1 describirá las distintas dimensiones en las que se divide una organización. La sección 2.2 describirá los lenguajes de modelado organizativo (OML), una forma de modelar OCMAS. La sección 2.2.1 presentará el Virtual Organization Model (VOM), un OML desarrollado dentro de nuestro grupo de investigación, que servirá de base para algunas de las propuestas de la presente tesis de máster. Otra forma de modelar un OCMAS es mediante una formalización, que se presentarán en la sección 2.3. Además, se muestran dos elementos utilizados para mejorar la coordinación de los agentes que pueblan un OCMAS. La sección 2.4 describe los artefactos, elementos de un SMA que los agentes usan para conseguir sus objetivos. Por otro lado, los mecanismos organizativos, descritos en la sección 2.5 son una propuesta importante a la hora de modificar el comportamiento de los agentes para aumentar la cooperación entre ellos. Por último, la sección 2.6 dará una visión de las propuestas más relevantes en lo referido a adaptación en sistemas multiagente centrados en la organización.

## 2.1. Dimensiones de la organización

A la hora de modelar una organización, se deben considerar diversas dimensiones. Estas dimensiones se pueden dividir en: (i) *estructural*, que describe las entidades que pueblan el sistema; (ii) *funcional*, que detalla las funciones, objetivos y servicios de la organización; (iii) *dinámica*, que considera las interacciones entre los elementos y sus efectos; (iv) de *entorno*, en el que se describen los elementos que rodean al sistema; y (v) *normativa*, que define los mecanismos usados por la sociedad para influenciar el comportamiento de sus miembros y se aplica sólo a nivel de sistema y de agente.

La **Dimensión Estructural** [AC09] comprende todos los elementos de

la organización que son independientes de los agentes que forman parte de ella. En consecuencia, se basa en roles, grupos y sus patrones de interacción. Los conceptos habituales que se emplean en la *Dimensión Estructural* son:

- Un *rol* representa una descripción abstracta del comportamiento de un miembro de la organización.
- Un *grupo* especifica el contexto o la localización de las actividades de los agentes.
- Es posible especificar diferentes *tipos de dependencias* entre roles, como la herencia de roles, la compatibilidad, la comunicación, la coordinación, la autoridad, el poder o el control.
- La *topología* de una organización predefine un conjunto de dependencias y grupos que serán necesarios para el correcto funcionamiento de la organización. Así, determina los patrones de interacción entre miembros. Son ejemplos de topologías: jerarquía, matricial, federación, etc. [Arg08]
- Una *sociedad* representa una colección estructurada de entidades que persiguen una meta común.
- Un *agente* es una entidad (humano o agente) con manifestaciones físicas concretas, que juega roles dentro de la organización.

La **Dimensión Funcional** [AC09] especifica los objetivos globales de la organización, las funciones y servicios ofrecidos, los objetivos perseguidos por diversos componentes de la organización y las tareas y planes que deben seguir para alcanzar estas metas.

Los servicios representan una funcionalidad que los agentes proporcionan a otras entidades, independientemente del agente concreto que los usa. Sus características principales son [AC09]: (i) sincronización, que implica una interacción entre entidades que ofrecen el servicio y aquellas que requieren su uso; (ii) publicación, con lo que el servicio se registra en un directorio de servicios y otras entidades pueden encontrarlo; (iii) participación, por ejemplo, las entidades que consumen el servicio pueden cambiar en el tiempo; (iv) estandarización de entidades, ya que los consumidores y proveedores de un servicio están relacionados con roles específicos, para los cuales las restricciones se definen mediante normas, permisos de acceso a los recursos, etc.; (v) estandarización de la funcionalidad, ya que los servicios se describen en base a entradas, salidas, precondiciones y postcondiciones, haciendo más fácil la descripción de su funcionalidad; (vi) tangibilidad, ya que los servicios

suelen producir productos tangibles que pueden ser empleados para evaluar tanto la calidad y la eficiencia del servicio como la satisfacción del cliente; y (vii) coste, por ejemplo, la producción del servicio y su consumo implica algunos costes y/o beneficios.

La *Dimensión Funcional* está conectada con la *Dimensión Estructural* a través del concepto de rol, ya que un rol describe una posición en un sistema que tiene una funcionalidad asignada; y con la *Dimensión Dinámica* por medio del concepto de servicio, ya que implica interacciones entre entidades.

Los conceptos típicos que se usan en la *Dimensión Funcional* son [AC09]:

- Una *tarea* representa la forma de hacer algo para satisfacer una meta.
- Un *servicio* o *actividad* representa la funcionalidad que los agentes ofrecen a otras entidades. Además, implica interacciones entre entidades. Los servicios pueden ser atómicos (tarea simple) o formados por numerosas tareas.
- Una *meta* representa situaciones que los agentes (o roles) desean llevar a cabo. Pueden dividirse en *softgoals* [BGG<sup>+</sup>04] (requisitos no funcionales, útiles para modelar cualidades del software tales como la seguridad, la actuación o el mantenimiento) y *hardgoals* (metas operativas, por ejemplo, resultados medibles y específicos que los agentes esperan conseguir).

La **Dimensión Dinámica** especifica cómo la organización evoluciona con el tiempo, detallando la forma en la que los agentes entran y dejan la organización, cómo adoptan ciertos roles de acuerdo a sus capacidades y habilidades, y cómo pueden participar en las unidades o grupos de la organización en las que son admitidos.

Esta dimensión también detalla las interacciones que tienen lugar entre los miembros de la organización y las entidades externas. Las interacciones consisten en múltiples entidades independientes que se coordinan ellas mismas para alcanzar sus metas individuales y conjuntas [Lin00].

Los conceptos típicos empleados en la Dimensión Dinámica son [AC09]:

- Una *comunicación* es una interacción entre dos agentes y está compuesta de uno o más mensajes. La información intercambiada durante una comunicación se compone de conceptos, predicados o acciones definidas en la ontología.
- Un *protocolo de comunicación* restringe el flujo de mensajes y la semántica de cada mensaje.

- La *promulgación de un rol* implica la asignación final de un rol a un agente.

La **Dimensión del Entorno** describe cómo los agentes están conectados a otros tipos de entidades como los objetos, aplicaciones o recursos; cómo estos elementos están representados conceptualmente, por medio de una ontología; y cómo los agentes pueden percibir y actuar sobre el entorno.

Los conceptos típicos que se usan en la *Dimensión del Entorno* son [AC09]:

- Un *recurso* representa objetos del entorno que no proporcionan una funcionalidad específica, pero que resultan indispensables para la ejecución de tareas. Pueden ser o no consumibles, tener un estado inicial, un umbral inferior y superior y una granularidad.
- Una *aplicación* representa interfaces funcionales que se describen con un nombre, varios parámetros, precondiciones, postcondiciones y resultados.
- Un *actor* representa una entidad con metas estratégicas e intencionalidad, que puede ser un agente físico (por ejemplo, una persona, un animal o un coche), o un agente software.
- Un *artefacto* es una entidad de diseño pasiva, dinámica y que posee un estado, para encapsular y proporcionar algún tipo de función.
- Las *ontologías* son modelos conceptuales que encarnan conceptualizaciones compartidas de un dominio dado [Gru93]. Los lenguajes de las ontologías se usan por aplicaciones (máquinas) que necesitan procesar el contenido de la información en vez de sólo presentar la información a los humanos.

Finalmente, la **Dimensión Normativa** [AC09] determina el conjunto de reglas y acciones definidas para controlar el comportamiento de los miembros de la organización. Las normas se han usado extensamente como mecanismos para limitar la autonomía dentro de las sociedades, para resolver problemas de coordinación, especialmente cuando no se puede ejercer control social total y directo. La *Dimensión Normativa* define un mecanismo de coordinación que intenta: (i) promover los comportamientos satisfactorios para la organización, por ejemplo las acciones que contribuyen a alcanzar las metas globales; y (ii) evitar acciones perjudiciales, por ejemplo acciones que llevan al sistema a ser insatisfactorio o inestable. Por tanto, la definición normativa permite especificar el comportamiento deseado en base a las acciones que los agentes deben

hacer o no. De forma más específica, las normas determinan la funcionalidad de los roles restringiendo sus acciones y estableciendo sus consecuencias.

La *Dimensión Normativa* está muy relacionada con el resto de dimensiones a través del concepto de norma, debido a que diversas normas o políticas pueden ser definidas para controlar las relaciones estructurales de los elementos del sistema, sus comportamientos esperados, la asignación de roles a los agentes y también los permisos de acceso al entorno. En muchos trabajos no se ha llevado a cabo una distinción clara de la dimensión normativa. Por ejemplo, las normas o los derechos no han sido definidos explícitamente, pero se asumen de manera implícita en otras dimensiones. Por ejemplo, las dependencias de roles de la Dimensión Estructural podrían ser consideradas como normas indirectas que restringen los tipos de roles que los agentes pueden jugar.

Los elementos típicos de las normas que se emplean en la *Dimensión Normativa* son [AC09]:

- Las *condiciones de activación* son los estados del entorno que causan que una norma se active. El *plazo* establece la condición que debe cumplir una norma para estar activa.
- Una *obligación* es una norma que ocasiona penalizaciones si no se cumple. Por el contrario, una *prohibición* implica una sanción cuando se cumple la acción prohibida. Un *permiso* es un tipo de política que prescribe las acciones y estados permitidos de un agente y/o recurso.
- Una *sanción* implica una acción o hecho que no es deseable para los agentes a los que se dirige la norma. Una *recompensa* representa el reconocimiento por un servicio, esfuerzo o logro.
- Un *contrato* representa los beneficios (derechos) y obligaciones de sus participantes.

Esta sección ha mostrado las diferentes dimensiones en las que se divide una organización, dando una visión de los elementos y relaciones entre ellos que estarán en cada una. Como se podrá comprobar en la siguiente sección, los diseñadores de OML no siempre contemplan por completo estas dimensiones a la hora de presentar sus lenguajes de modelado, sino que se centran en ciertos aspectos, dejando de lado en determinadas ocasiones otros elementos. Por ello, la siguiente sección dará además una visión de las dimensiones y elementos de una organización que tratan distintos OML.

## 2.2. Lenguajes de Modelado de Organizaciones

El metamodelado es un mecanismo que permite definir lenguajes de modelado de manera formal, estableciendo las primitivas y las propiedades sintáctico-semánticas de un modelo [vG91]. Un metamodelo organizativo describe las entidades y relaciones que comprenden una organización y su entorno.

Los Lenguajes de Modelado de Organizaciones (Organizational Modeling Language, OML) son usados por diseñadores de OCMAS para definir los elementos que deberá contener el sistema en tiempo de ejecución. Los OML en su mayoría extienden lenguajes de modelado existentes para permitir introducir elementos organizativos.

Las organizaciones de agentes se consideran abiertas, con lo que los agentes que las poblarán serán heterogéneos. Los OML se usan para modelar la coordinación de los agentes dentro de estos sistemas abiertos, además de para establecer mecanismos que controlen la organización a nivel social [DMWD02]. Estos modelos incluyen perspectivas tanto a nivel individual como a nivel organizativo, y algunas propuestas permiten adaptar los modelos de forma dinámica para dar respuesta a cambios en la estructura organizativa o del entorno [DD06a].

Hasta el momento se han propuesto varios modelos de organización, algunos de ellos dando soporte a las metodologías de desarrollo de SMA, que se han enumerado en distintos artículos de revisión de estado del arte [CSB05, OB07, ABD<sup>+</sup>07, ABFF<sup>+</sup>10]. En la figura 2.1 se puede comprobar la evolución en el tiempo que han experimentado los distintos lenguajes de modelado (y/o sus metodologías asociadas). En los últimos años han surgido diversos metamodelos que, o bien suponen una evolución de algún OML existente, o presentan nuevas aproximaciones al modelado de SMA. Además, varias propuestas, como MEnSA [ABC<sup>+</sup>08], toman elementos de distintos metamodelos para definir el suyo propio.

Uno de los primeros modelos de organizaciones de agentes fue propuesto por Ferber y Gutknecht en **AAALADIN** [FG98] y extendido en **AGR** [FGM03]. Sus conceptos básicos de agentes, grupos y roles pueden encontrarse en prácticamente todos los modelos de forma similar. El modelo de **Gaia** [ZJW03] da soporte a la metodología Gaia, una de las metodologías SMA más extendidas. Este modelo considera también la organización en términos de agentes, grupos y roles, que tienen responsabilidades, permisos, actividades y protocolos.

Otro modelo relevante pionero es el framework **ISLANDER** [ERS<sup>+</sup>01], que toma en cuenta los aspectos o dimensiones estructurales, de interacción y normativos de una organización. Por su parte, el modelo de **OperA** [Dig03]

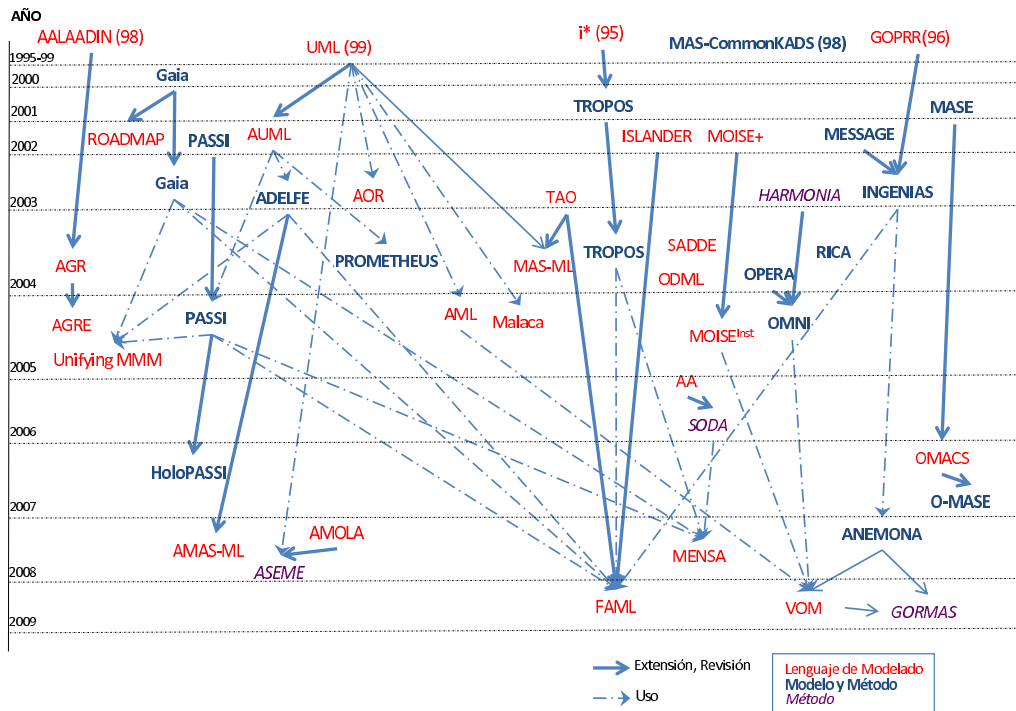


Figura 2.1: Evolución temporal de los OML

permite a agentes heterogéneos entrar en la organización con sus propias metas, creencias y capacidades, sin asumir que estos agentes sean cooperativos.

Existen diversas aproximaciones metodológicas, como las proporcionadas en Tropos, PASSI, SODA, MENSA y OMACS. **Tropos** [BGG+04] mejora principalmente el análisis de requisitos del sistema, por medio de actores, softgoals, planes y dependencias. En **PASSI** [Cos05], los conceptos del metamodelo se dividen en aspectos de dominio de problema, agencia y solución, por lo que pueden ser refinados en conceptos más abstractos, relacionados con las fases de requisitos y análisis (dominio del problema) hacia conceptos más prácticos que pueden ser implementados en plataformas multiagente específicas (dominio de solución).

La tabla 2.1 compara distintos OML. Entre ellos se encuentran metamodelos surgidos en los últimos años como PIM4Agents, OMACS, NormML, MENSA, FAML y VOM; además de modelos relevantes como SODA e IN-



GENIAS. El objetivo de esta tabla es contrastar los elementos más representativos de una organización de agentes que presentan estos OML.

Dos de los conceptos más importantes para la presente tesis de máster fueron introducidos en el OML de la metodología **SODA**, los *artefactos* y los *espacios de trabajo* (que serán descritos en la sección 2.4), que introducen una nueva aproximación en el modelado del entorno y la topología de un OC-MAS. En SODA, los elementos del metamodelo se van identificando según las distintas fases de la metodología asociada, llamada también SODA. La primera fase es la de *análisis*, que comprende los pasos de *análisis de requisitos*, en el que se identifican los *actores*, *requisitos* y los *legacy systems* que rodean al sistema; y de *análisis*, donde se define la *topología*, las *tareas* y las *funciones* del sistema. La segunda parte es la fase de *diseño*, que se divide en dos tareas: *diseño arquitectónico* que define los *roles*, que se asocian a *tareas*, y las dependencias entre roles se convierten en *interacciones*. En el último paso, el *diseño detallado*, la topología se refina en los *espacios de trabajo* y se definen los *agentes*, las *sociedades de agentes* y los *artefactos*.

SODA toma conceptos de las dimensiones estructural, dinámica y del entorno, sin modelar por tanto conceptos como normas u objetivos.

Otra metodología cuyo metamodelo repite su mismo nombre es **INGENIAS** [PG03], que posee un lenguaje de modelado de organizaciones que se descompone en los siguientes puntos de vista:

1. **Punto de vista de la organización.** Describe cómo se agrupan los *componentes del sistema* (agentes, roles, recursos y aplicaciones), qué *tareas* ejecutan en común, qué *objetivos* comparten y qué *restricciones* existen en la interacción entre agentes. Estas restricciones se expresan en forma de relaciones de subordinación y cliente-servidor.
2. **Punto de vista del agente.** Describe *agentes únicos*, sus *tareas*, *objetivos*, *estado mental* inicial y *roles* jugados. Además, los modelos de agente se usan para describir estados intermedios de los agentes. Estos estados se presentan usando *objetivos*, *hechos*, *tareas* o cualquier otra entidad del sistema que ayude a describir este estado. De esta forma, un modelo de agente podría representar en qué estado debe encontrarse un agente para empezar una interacción.
3. **Punto de vista de interacción.** Describe cómo se llevan a cabo las *interacciones* entre agentes. Cada declaración de interacción incluye los agentes implicados, los objetivos perseguidos por la interacción y la descripción del protocolo que sigue la interacción.
4. **Punto de vista de tareas y objetivos.** Describe las relaciones entre *objetivos* y *tareas* y las estructuras de tareas y objetivos. También se

usa para expresar las entradas y salidas de las tareas, y sus efectos en el entorno o en el estado mental del agente.

5. **Punto de vista de entorno.** Define la *percepción* de un agente en base a elementos existentes del sistema. También identifica los *recursos* del sistema y los *responsables* de su control.

En INGENIAS se tienen en cuenta las dimensiones estructural, funcional, dinámica y del entorno, sin entrar en aspectos normativos.

Hahn *et al.* definen **PIM4Agents** [HMMF09], un metamodelo independiente de la plataforma para el diseño de SMA que se puede usar para modelar un sistema de agentes de manera abstracta sin centrarse en los requisitos concretos de una plataforma. Los modelos independientes de la plataforma que se generan podrán después ser transformados a código ejecutable por plataformas orientadas al agente como JADE [BPR99] y JACK [HRHL01]. Este metamodelo se estructura en las siguientes vistas:

1. **Vista Multi-Agente:** contiene los cimientos de un SMA, incluyendo conceptos como *Agente*, *Cooperación*, *Capacidad*, *Interacción*, *Rol* o *Entorno*.
2. **Vista de agente:** describe *entidades autónomas* aisladas, las *capacidades* que poseen para solucionar tareas y los *roles* que juegan dentro del SMA.
3. **Vista de comportamiento:** describe cómo los *planes* se componen de estructuras de control complejas y tareas atómicas simples como el envío de mensajes y cómo se construyen los flujos de información.
4. **Vista de organización:** describe cómo las entidades autónomas simples cooperan dentro del SMA y cómo se pueden definir las *estructuras organizativas*.
5. **Vista de rol:** cubre las representaciones abstractas de posiciones funcionales de entidades autónomas dentro de una organización u otras relaciones sociales.
6. **Vista de interacción:** describe cómo ocurren las *interacciones*, en forma de protocolos de interacción, entre entidades autónomas u organizaciones.
7. **Vista de entorno:** contiene cualquier tipo de *recurso* creado, compartido o usado dinámicamente por los agentes o las organizaciones, respectivamente.

PIM4Agents presenta conceptos de las dimensiones estructural, funcional, dinámica y del entorno de una organización de agentes.

El modelo **OMACS** [DeL09], procedente del metamodelo MaSE [DeL04], a su vez basado en AGR [FGM03], proporciona los mecanismos adecuados para permitir al sistema reorganizarse en tiempo de ejecución y adaptarse a su entorno y a las capacidades cambiantes de sus agentes. Sus principales elementos son las *metas*, *roles*, *agentes* y *capacidades*, además del *entorno*, en el que se encontrarán los *objetos del entorno*, y sus *relaciones*. Este metamodelo está soportado por la metodología O-MaSE [GODOV08]. El metamodelo de O-MaSE incluye además de los conceptos de OMACS, otros conceptos como *actores externos*, usados para representar cada agente, sistema o usuario que se encuentra fuera del sistema que se está diseñando; *protocolos*, que definen un conjunto de mensajes permitidos que pueden existir entre actores externos y una organización (protocolos externos) o entre agentes dentro de una organización (protocolos internos); *planes*, que son un tipo de capacidad de agente que define una secuencia de acciones o mensajes que se pueden usar para alcanzar la meta de un agente. Por último, el modelo de domino de O-MaSE consiste en un conjunto de *objetos del entorno* que describen los objetos en el dominio del sistema y un conjunto de *propiedades* que especifican los principios y procesos que rigen el entorno. Este modelo toma en cuenta las dimensiones estructural, funcional y de entorno.

También han surgido OMLs para dar soporte a propósitos más específicos, como es el caso de **NormML** [dSBF10], surgido a partir de SecureUML [BDL04], que es un OML para modelar *roles*, *permisos*, *acciones*, *recursos* y *restricciones de autorización*. NormML extiende SecureUML con el concepto de *norma*, introduciendo por tanto uno de los elementos principales cuando se trata de definir una organización de agentes. Estas normas describen el conjunto de acciones que el agente está obligado, las que se le permite llevar a cabo y las que tiene prohibidas. Estas normas pueden estar asociadas tanto a agentes individuales como a roles. Los elementos básicos del metamodelo son las normas, agentes y acciones de agentes (que a su vez se extienden en *atributo*, *método*, *entidad* y *asociación*). De las normas se puede especificar el periodo de tiempo en el que se encuentran activas, junto a sus condiciones de activación y desactivación, permitiendo especificar de forma completa el modelo de una norma. Las dimensiones estructural, funcional, del entorno y normativa quedan representadas en este OML.

El proyecto **MEnSA** [ABC<sup>+</sup>08] tiene como objetivo extraer elementos de procesos y OMLs para el desarrollo de SMA para formar un único metamodelo con su metodología asociada. El metamodelo de MEnSA [ABC<sup>+</sup>08] integra conceptos de los modelos de Tropos, Gaia, SODA y PASSI y también se divide en las fases de requisitos, diseño e implementación. La fase de

*requisitos* de MEnSA está formada principalmente por conceptos extraídos de Tropos y Gaia, además de conceptos relacionados con el entorno extraídos de SODA. La figura central de este modelo es el *agente de requisitos*, que juega uno o más *roles*. Estos agentes forman la *organización*, que tiene sus reglas organizativas. Cada rol es responsable del cumplimiento de uno o más requisitos, que a su vez pueden especificarse en metas o en requisitos no funcionales. Los roles, además, poseen relaciones de dependencia y permisos entre ellos. Finalmente, la fase de requisitos define la topología del sistema.

Por su parte, en la fase de *diseño* se usa el concepto de *agente de diseño*, extraído de PASSI, que posee un conjunto de comunicaciones, definidas por un *protocolo*. En esta parte se encuentran definidos los *servicios* y la topología se diseña a través de los *espacios de trabajo*, que estructuran el entorno. En el entorno se encontrarán los *artefactos*, entidades usadas por los agentes para alcanzar sus metas, extraídas del metamodelo de SODA. Los agentes de diseño y los artefactos se agrupan en composiciones, que podrán ser sociedades (con comportamiento proactivo) o agregaciones (con comportamiento reactivo). Por último, el concepto de *regla* es importante, ya que restringe los comportamientos de las entidades del sistema.

MEnSA toma conceptos de las dimensiones estructural, funcional, dinámica y del entorno de la organización.

De manera similar a MEnSA, **FAML** [BLHS<sup>+</sup>09] presenta un metamodelo que incorpora los elementos más destacables de diversos OMLs. FAML tiene dos capas, la capa de *diseño*, cuyo concepto principal es el sistema, y la capa de *ejecución*, con el entorno como elemento principal. Además, en FAML existe la visión interna al agente y la visión externa al agente. Por tanto, el metamodelo de FAML se compone de cuatro niveles, surgidos de la composición de ambos conceptos: (i) *nivel de sistema*, en tiempo de diseño y externo al agente; (ii) *nivel de definición de agente*, en tiempo de diseño e interno; (iii) *nivel de entorno*, a nivel de ejecución y externo al agente; y (iv) *nivel de agente*, en tiempo de ejecución y externo al agente. FAML separa de manera clara los conceptos a nivel de diseño como de nivel de ejecución. El nivel de diseño contempla conceptos como el *sistema*, el producto final del desarrollo; la *definición de la organización*; los *roles*, la *definición de los agentes*, incluyendo sus patrones de comportamiento, además de sus compatibilidades, dependencias y relaciones; los *servicios*, los *objetivos*, el *entorno* y la *ontología*. Por otra parte, el modelo de ejecución está formado en su mayoría por *instancias* de estos elementos, como los *agentes*, los *recursos* y los *roles*. Además, incluye elementos dinámicos, como el *entorno*, ya que se almacenan los eventos que han ocurrido en él durante la ejecución del sistema.

Este OML surge con la intención de ser modificado con el tiempo, incor-

porando características que otros metamodelos aportan, pero que FAML no tiene incorporados. Es por eso que, después de definir el metamodelo, FAML debe pasar por una fase de validación contra otro u otros metamodelos. Hasta este momento, FAML ha sido contrastado con TAO, Islander, Adelfe, PASSI, Gaia, INGENIAS y Tropos, incorporando los conceptos más importantes de cada uno de estos OMLs.

Al tomar conceptos de diferentes OMLs, FAML tiene conceptos de todas las dimensiones de una organización de agentes.

Por último, el Virtual Organization Model (**VOM**) [CAJB09] es un meta-modelo que permite la definición de organizaciones virtuales. Desarrollado en el seno de nuestro grupo de investigación, este modelo servirá como base para integrar en él nuestro trabajo sobre artefactos de mecanismos organizativos. El VOM se describirá en profundidad en la sección 2.2.1.

Conceptos organizativos								
	SODA	INGENIAS	PIM4Agents	OMACS	NormML	MEEnSA	FAML	VOM
<b>Dimensión estructural</b>								
<b>Roles</b>	✓	✓	✓	✓	✓	✓	✓	✓
<b>Grupos</b>		✓	✓	✓		✓	✓	✓
<b>Agentes</b>	✓	✓	✓	✓	✓	✓	✓	✓
<b>Relaciones</b>		✓	✓		✓	✓	✓	✓
<b>Topología</b>	✓	✓				✓	✓	✓
<b>Dimensión funcional</b>								
<b>Capacidades</b>			✓	✓	✓			
<b>Servicio</b>		✓				✓	✓	✓
<b>Objetivos</b>		✓				✓	✓	✓
<b>Dimensión dinámica</b>								
<b>Interacciones</b>	✓	✓	✓			✓	✓	✓
<b>Transiciones</b>							✓	
<b>Dimensión del entorno</b>								
<b>Entorno</b>	✓		✓	✓		✓	✓	✓
<b>Recurso</b>	✓	✓	✓	✓	✓	✓	✓	✓
<b>Dimensión normativa</b>								
<b>Normas</b>					✓		✓	✓

Tabla 2.1: Comparación entre distintos OML

Tras observar la tabla 2.1 se comprueba que los metamodelos surgidos a partir de la unión de elementos de distintos OML, como son MEEnSA y FAML, contienen la mayoría de elementos analizados. También VOM toma en cuenta la mayor parte de conceptos relevantes para los OCMAS. Por otro lado, se observa que los OMLs más clásicos como SODA, o los que están centrados en una funcionalidad concreta, como NormML, poseen menos características que el resto. Esto se produce por dos razones. La primera es que los modelos más antiguos no presentan ciertos conceptos debido a que no se contemplaban en el estado del arte en el momento en que se desarrollaron estos modelos. De todas formas, existen OMLs que han adaptado nuevos conceptos, como SODA, que adoptó el concepto de artefacto. La segunda, que los OML que se

centran en una funcionalidad determinada no necesitarán todos los conceptos, y su inclusión en el metamodelo hará aumentar la complejidad a la que se tienen que enfrentar los diseñadores.

El siguiente apartado da una visión más detallada del VOM, que en la sección 4 será extendido con nuestras propuestas, centradas principalmente en la introducción de los artefactos de mecanismos organizativos.

### 2.2.1. Virtual Organization Model

Esta sección describe el Virtual Organization Model (VOM), un OML centrado en la definición de organizaciones virtuales, desarrollado en nuestro grupo de investigación. Este OML se compone de cinco dimensiones: funcional, estructural, dinámica, de entorno y normativa.

- **Dimensión Estructural.** Describe los componentes del sistema y sus relaciones. En ella se definirá la organización, compuesta de agentes y unidades organizativas, los roles, las relaciones sociales y las relaciones 'contiene' entre unidades organizativas y otras entidades de la organización.
- **Dimensión Funcional.** Detalla la funcionalidad del sistema basándose en servicios, tareas y objetivos. En esta dimensión quedan definidos los objetivos, los grupos de interés que interactúan con las unidades organizativas, los servicios ofrecidos por la organización, los recursos usados por la organización y las relaciones sobre requisitos de servicios y recursos.
- **Dimensión Dinámica.** Define las interacciones entre agentes, además de los estados mentales del sistema y el proceso de promulgación de roles, definiendo los roles que una unidad organizativa o un agente pueden jugar.
- **Dimensión del entorno.** El entorno de la organización se define mediante los recursos y aplicaciones que la organización tiene disponibles. En esta dimensión se definen además los puertos, entidades que controlan el acceso a servicios y productos del entorno.
- **Dimensión normativa.** Describe las restricciones normativas sobre el comportamiento de las entidades del sistema, incluyendo normas organizativas que los agentes deben cumplir, con sanciones y recompensas asociadas.

La figura 2.2 muestra una versión simplificada del metamodelo VOM, representando sus principales entidades y relaciones. En esta figura es posible identificar las cinco dimensiones que componen el metamodelo VOM. Se muestran las entidades y relaciones más representativas del metamodelo, que han sido seleccionadas para proporcionar una visión general de las entidades del metamodelo VOM y las relaciones que definen cada dimensión. Estos elementos se representan alrededor de la entidad *Unidad Organizativa*. Rodeándola, las cinco dimensiones que componen el metamodelo se muestran usando diferentes tramas (ver leyenda en la figura 2.2).

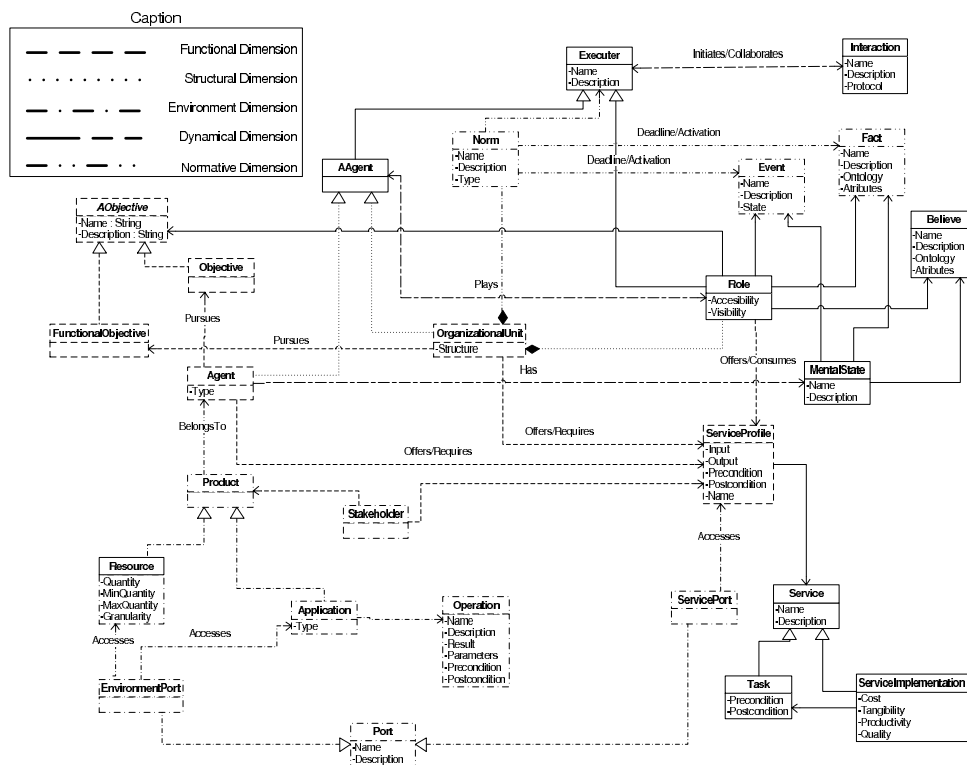


Figura 2.2: Representación esquemática del metamodelo VOM

En esta vista simplificada, de la *Dimensión Estructural* se han mostrado las unidades organizativas, roles y agentes, además de la relación 'contains' entre roles y *Unidad Organizativa*. Este metamodelo simplificado señala para la *Dimensión Funcional* las relaciones que la *Unidad Organizativa* mantiene con objetivos y servicios. Además, representa los productos y servicios ofrecidos o requeridos por los grupos de interés. De la *Dimensión Dinámica* se definen las interacciones entre agentes, además de los estados mentales del sistema. La *Dimensión del entorno* queda representada por medio de

los recursos, aplicaciones y puertos que rodean al sistema. Por último, de la *Dimensión Normativa* se muestran las normas.

Las organizaciones se estructuran por medio de Unidades Organizativas (OU, por sus siglas en inglés), que representan un conjunto de agentes que llevan a cabo algunas actividades o tareas específicas y diferenciadas, siguiendo un patrón predefinido de cooperación y comunicación. Una OU está formada por diferentes entidades durante su ciclo de vida, que pueden ser agentes o unidades organizativas, vistas como una única entidad.

Las entidades del sistema son capaces de ofrecer o solicitar servicios y su comportamiento está motivado por los objetivos que persigue. Los servicios representan la funcionalidad que los agentes ofrecen a otras entidades, independientemente del agente concreto que hace uso de ella. Por otra parte, una unidad organizativa puede también publicar los servicios que requiere, con lo que los agentes externos pueden decidir si entrar a participar dentro de ella, proporcionando estos servicios.

### 2.3. Formalización

Una de las aproximaciones para modelar OCMAS es la de la formalización. Basadas en distintas lógicas y métodos formales, se han realizado distintas propuestas para el modelado de OCMAS, cada una de ellas aportando su visión particular y adaptando sus formalizaciones a los distintos tipos de sistemas que se buscan construir. En esta sección se presentan las características que las formalizaciones deben cumplir, y se describirá un conjunto de las mismas que se ha considerado relevante, en el que se encuentran: OperA [Dig03], LAO [DD07], Process-Oriented Modeling Framework (POMF) [PS06], MOISE<sup>Inst</sup> [GBKD05], MACODO [HWHJ09], PopOrg [dRCD08] y las formalizaciones propuestas por Grossi [GDD<sup>+</sup>07] y Jonker [JSTY07].

En [DD06b] se enumeran las características que deben cumplir los modelos formales de organizaciones que quieran ser capaces de poner en práctica situaciones realistas. En la tabla 2.2 se muestra la presencia de los elementos más relevantes en las distintas formalizaciones analizadas en esta tesis de máster. Las características a considerar por los modelos formales son:

1. Representar nociones de habilidad y actividad de agentes, sin necesidad de conocimiento sobre las acciones específicas disponibles para cada agente.
2. Aceptar las limitaciones de las capacidades de los agentes.
3. Representar la habilidad y actividad de un grupo de agentes.



4. Hacer frente a cuestiones temporales, en especial al hecho de que la actividad lleva tiempo.
5. Representar el concepto de 'ser responsable' por la consecución de un estado dado de los acontecimientos.
6. Representar objetivos organizativos y su relación con la actividad de los agentes, relacionando actividad y estructura organizativa.
7. Hacer frente a las limitaciones de los recursos y la dependencia de la actividad en los recursos.
8. Representar la dinámica de la organización, definiendo cómo evoluciona en el tiempo y representando las condiciones en la población.
9. Representar organizaciones en términos de roles organizativos o posiciones.
10. Ocuparse de los aspectos normativos, que implican límites para las acciones y definen qué hacer frente a las violaciones de las normas.

Virginia Dignum presentó en su tesis doctoral un marco de trabajo formal para definir organizaciones llamado OperA [Dig03]. Este trabajo permite definir roles, incluyendo sus objetivos, derechos y normas asociadas; grupos, formados por conjuntos organizados de roles que contienen normas comunes; relaciones de dependencia entre roles (que pueden ser jerárquicas, de mercado o una red); y normas. OperA también posee una estructura para definir las interacciones entre agentes. Las actividades se articulan mediante escenas, que siguen guiones (*scripts*). Para representarlas se utilizan *landmarks*, basados en los modelos de Kripke [Smo73] y similares a la representación de las Instituciones Electrónicas [Est03].

OperA también gestiona la comunicación, usando lógica ilocutoria [SV85] entre los agentes. El concepto básico de esta lógica es el acto ilocutorio, que consiste en tres partes: (i) *contenido proposicional*, que describe la temática del acto de habla; (ii) *contexto ilocutorio*, que indica el conocimiento relevante sobre la situación en la que se lleva a cabo el acto de habla (emisor, receptor, tiempo, lugar, etc.); y (iii) *fuerza ilocutoria*, que determina las razones y el propósito de la comunicación. Por último, OperA es capaz de controlar las interacciones entre agentes mediante contratos (entre un par de agentes o entre un agente y su entorno). Los contratos se contemplan como entidades dinámicas que pueden ser negociadas, aceptadas, ejecutadas y cumplidas.

Para modelar las dependencias entre roles OperA ofrece tres relaciones: de *poder* (*power*), *autorización* (*authorization*) y *caridad* (*charity*). La relación

de poder expresa que un agente es capaz de pedir a un agente subordinado que alcance sus objetivos. La relación de autorización permite a un agente dar autorización a otro para que realice una acción determinada. Por último, la relación de caridad permite a un agente pedir permiso para realizar una acción. Si se acepta este permiso, aparece una obligación. Además, en OperA, el objetivo que un agente subordinado puede tomar de su agente superior queda determinado por el tipo de relación existente entre roles, y tiene tres posibilidades: (i) relación *jerárquica*, donde el agente superior delega un objetivo al agente subordinado; (ii) relación de *mercado*, donde el agente subordinado puja por un objetivo al agente superior; y (iii) relación de *red*, donde tanto el agente subordinado como el superior pueden solicitarse un objetivo mutuamente. La jerarquía entre los roles de la organización se establece usando estas relaciones.

Virginia y Frank Dignum presentaron la llamada *lógica para organizaciones de agentes* (LAO) [DD07] que es una extensión de la lógica CTL [Eme91]. Esta lógica es capaz de expresar la estructura y las estrategias de una organización. En este trabajo el concepto de rol se expresa a través de las capacidades, habilidades, actividades e intentos de los agentes y grupos que componen la organización. La capacidad de un agente se define como el conjunto de proposiciones atómicas que el agente puede cumplir. Las habilidades de un agente expresan el potencial que un agente tiene para actuar sobre el mundo. Las actividades quedan representadas como las acciones que el agente ejecuta. Finalmente, los intentos representan la incertidumbre del sistema, ya que un agente puede tener la capacidad de ejecutar una acción, pero esa acción puede no tener éxito siempre. Esta formalización también define diferentes estados del mundo en el que se encuentra el sistema y sus transiciones, y permite establecer la topología del SMA, siendo los principales tipos de estructuras las redes y las jerarquías.

Popova y Sharpanskykh presentaron en [PS06] un marco de trabajo formal *orientado a procesos*, denominado *Process Oriented Modeling Framework* (POMF). Este trabajo formal está expresado en el lenguaje formal lógico de predicados  $L_{PR}$  [Man96] que incluye técnicas de análisis dedicado para comprobar la corrección de los modelos. Este trabajo se estructura en cuatro vistas: (i) vista orientada al proceso, (ii) vista orientada a la organización, (iii) vista de agente y (iv) vista orientada a la actuación. La vista principal es la *orientada al proceso*, en la que se encuentran definidas las tareas (incluyendo su nombre, intervalo de duración y descripción), procesos (definidos usando una tarea como patrón), flujos de trabajo (definidos como un conjunto de procesos ordenados temporalmente) y recursos (cuya definición incluye su nombre, categoría, unidad de medida y tiempo de vida). La vista orientada al proceso se relaciona con la vista *orientada a la organización* por medio

de la entidad rol, que describe el conjunto de funcionalidades de los procesos organizativos en un flujo de trabajo determinado. Estas funcionalidades son después asignadas a individuos que las llevarán a cabo. Estos individuos son entidades del tipo agente, que se encuentran definidas en la *vista de agente*. Además, existe la *vista orientada a la actuación*, que describe las metas de la organización y los indicadores que ayudan a comprobar si se están cumpliendo. Este marco de trabajo también define una serie de restricciones: *estructurales*, que actuarán sobre el flujo de trabajo y las jerarquías de tareas y recursos; *restricciones genéricas del mundo físico*, que son independientes del dominio de la aplicación; y *específicas del dominio*, que pueden ser impuestas por la organización, procedentes del exterior o del mundo físico. Estas restricciones son verificadas para comprobar su corrección únicamente cuando se producen cambios en ellas (o sólo las más relevantes en el caso de las restricciones genéricas) lo que hace que el proceso de comprobación sea computacionalmente más eficiente que un algoritmo de *model checking* [Cla99].

Gateau *et al.* presentaron una formalización para OCMAS, denominada MOISE<sup>Inst</sup> [GBKD05] que se compone de cuatro especificaciones (*Funcional, Estructural, Normativa y Contextual*). La *Especificación Estructural* (SS) define los roles que jugarán los agentes del sistema, incluyendo las relaciones entre ellos. Además, la SS define un nivel estructural adicional denominado grupo, al que los roles pertenecen y donde se llevan a cabo las interacciones. La *Especificación Funcional* (FS) define las metas que el sistema debe alcanzar, usando *Esquemas Sociales*, que son árboles de descomposición de metas. La *Especificación Contextual* (CS) define los diferentes contextos que influyen en la dinámica de la organización y las transiciones entre ellos. Finalmente, la *Especificación Normativa* (NS) define los derechos y deberes de roles y grupos dentro de la organización. La unión de la SS, de la FS, de la CS y de la NS conforman la *Especificación Organizativa* (OS) del sistema. Todos los agentes que han adoptado un rol de la SS componen la Entidad Organizativa (OE), que es la parte del sistema que controla la dinámica de la organización, incluyendo agentes, grupos y esquemas (un conjunto de planes que el agente dentro de un grupo tiene que seguir para alcanzar sus objetivos).

El modelado de la estructura de una organización de MOISE<sup>Inst</sup> está heredado de MOISE<sup>+</sup>, donde Hübner *et al* [HSB02] presentan otra aproximación desde este punto de vista. Los autores modelan los tres conceptos principales de MOISE<sup>+</sup> (roles, relaciones de roles y grupos) por medio de tres niveles estructurales de una organización: (i) el *nivel individual (individual level)*, formado por los roles de la organización, y presenta relaciones de herencia entre roles; (ii) el *nivel social (social level)*, que se construye de relaciones de enlaces entre roles que se pueden clasificar en *conocimiento*, si los agentes

que juegan el rol origen están autorizados a tener una representación de los agentes que juegan los roles destino; *comunicación*, si los agentes son capaces de comunicarse entre ellos; y *autoridad* si los agentes fuente pueden tener autoridad en los agentes de destino; y (iii) el *nivel colectivo* (*collective level*), que define los grupos de agentes de la organización, estableciendo la *compatibilidad* entre roles y la *cardinalidad* de los roles.

La formalización de organizaciones propuesta por Grossi *et al.* [GDD<sup>+</sup>07] persigue representar la estructura organizativa, analizando tres dimensiones que ésta define de manera implícita: (i) *poder*, en el sentido de la delegación de actividades; (ii) *coordinación*, en relación con aspectos de conocimiento e información; y (iii) *control*, en relación con los asuntos de monitorización y recuperación. Las organizaciones se representan con las restricciones impuestas por estos tres aspectos. La estructura relaciona los roles de la organización ordenándolos con respecto a estas tres dimensiones. Los roles de la organización se conciben alrededor de tres nociones básicas: objetivos, normas e información. Los objetivos y normas expresan los compromisos y las restricciones de los agentes que toman un rol determinado, siendo además una herramienta de control de los agentes. En cuanto a la información, los agentes deben tener conocimiento del estado actual de la organización para saber si están jugando el rol de forma adecuada. Para definir la organización, los autores usan su propio lenguaje, *Org* [GDD<sup>+</sup>07], un lenguaje lógico proposicional multi-modal, cuya semántica está basada en modelos de Kripke.

Por su parte, Jonker *et al.* [JSTY07] definen un framework para el modelado y análisis formal de organizaciones, basadas en una representación genérica de las mismas a través de un conjunto de roles relacionados entre sí. Esta aproximación se apoya en un lenguaje de predicados ordenados. En este marco se permite definir tanto la *estructura* de la organización como su *dinámica*. El modelo de organización permite múltiples niveles de agregación. En el nivel más alto de organización, el sistema vendrá representado por un único rol, que se refinará en otros roles en los sucesivos niveles para aumentar el detalle en la representación de la organización. El entorno se define explícitamente en el modelo de organización. Además, la aproximación proporciona técnicas y herramientas formales para los diferentes tipos de análisis de modelos de organización.

Esta propuesta especifica las organizaciones como roles compuestos, que serán refinados en cada nivel de agregación en otros roles, simples o compuestos, que interactúan entre sí. El comportamiento de cada uno de estos roles se define mediante un conjunto de propiedades dinámicas, que se expresan usando un lenguaje lógico temporal, TTL [JT03]. Para modelar las normas de la organización, éstas se definen como propiedades estáticas y dinámicas del rol del nivel más alto de agregación. Finalmente, el entorno de la or-

ganización está poblado de agentes que, bajo determinadas circunstancias, pueden tomar un rol dentro de la organización.

Las organizaciones con conceptos auto-organizativos también han tenido aproximaciones formales para llevar a cabo su modelado. Este es el caso del marco de trabajo MACODO [HWHJ09], que se centra en la *dinámica de la organización*, usando Z como lenguaje de especificación formal [ASM80]. Para mantener el control de la organización en un sistema con una alta tasa de cambio se hace necesario el uso de un middleware que sea capaz de monitorizar y modelar la organización. Para realizar este modelado, MACODO cuenta con un modelo en el que se definen conceptos como roles, estableciendo además los conceptos de *puesto* (similar a una oferta de trabajo en una empresa) y *contrato* (un acuerdo entre un agente y una organización para un puesto determinado) de roles, para controlar los accesos a un rol disponible; agentes (incluyendo su contexto, consistente en el estado actual del entorno local del agente y su localización) y organizaciones. Para que un agente pueda entrar en una organización debe crear su propia organización y después unir ambas organizaciones en una sola, quedando el agente de esta forma integrado en la organización original. Para salir de la organización, el agente debe poner fin a todos los contratos que tiene activos en la organización. MACODO también es capaz de lidiar con la dinámica de la organización, incluyendo modificaciones en el contexto del sistema o en el conjunto de agentes del sistema y posee además reglas para mantener el correcto funcionamiento del sistema. Estas reglas pertenecerán a dos tipos. El primer tipo de reglas son las reglas que controlan la dinámica intra-organizativa, que describe cómo un agente puede entrar y salir de la organización de manera dinámica, un concepto muy importante cuando se habla de organizaciones abiertas. El segundo tipo de reglas son las referidas a la dinámica inter-organizativa, incluyendo la división y la unión de organizaciones.

Finalmente, para manejar la dinámica estructural de un SMA, da Rocha y Pereira presentan PopOrg [dRCD08], un modelo que se fundamenta en dos conceptos básicos: la *población* de la organización y su *estructura*. La población de un SMA es el conjunto de agentes que lo habitan, así como los comportamientos, acciones y procesos de intercambio (*exchange processes*) que los agentes pueden llevar a cabo. Cada proceso de intercambio viene determinado por una función que, para cada unidad de tiempo, especifica un par de acciones y determina una secuencia de intercambios que dos agentes ejecutarán para, de manera simultánea o intercalada, llevar a cabo estas acciones. Por su parte, la estructura de la organización se compone de roles, representando conjuntos de comportamiento; y los enlaces entre roles, que estarán restringidos según la capacidad que dos roles posean para relacionarse. Para relacionar la población y estructura, el sistema dispone de un tercer

elemento llamado implementación que relaciona los roles con los agentes y los enlaces (o servicios) con los *exchange processes*. Estas relaciones no siempre son uno a uno, con lo que un agente podría jugar distintos roles en la organización, varios agentes podrían tomar el mismo rol, varios enlaces podrían estar apoyados por distintos procesos de intercambio y distintos procesos de intercambio podrían dar soporte a un determinado enlace.

Este sistema es capaz de tratar tanto con la componente estática de la organización como con su dinámica, almacenando los distintos estados por los que el sistema va pasando durante su ejecución. Se contemplan diversos tipos de cambios en la organización, definidos según el elemento de la organización (población, estructura o implementación) al que afecten. Los cambios que se consideran en la población son: (i) cambios en el conjunto de comportamientos que un agente puede llevar a cabo; (ii) cambios en el conjunto de procesos de intercambio que dos agentes pueden ejecutar; y (iii) cambios en la población de agentes. Los tipos de cambio estructurales considerados son: (i) cambio en un rol; (ii) cambio en un enlace; (iii) cambio en el conjunto de roles; (iv) cambio en el conjunto de enlaces; y (v) cambio en el conjunto de enlaces que dos roles pueden establecer entre sí. Finalmente, los cambios en la implementación pueden darse: (i) en la manera en que se relacionan roles y agentes; y (ii) en la manera en la que se relacionan enlaces y procesos de intercambio. PopOrg posee el concepto de servicio (llamado *exchange value*), definido como una acción o tarea que un agente ejecuta, y que influye de manera positiva o negativa en otros agentes en su esfuerzo para conseguir un objetivo. Esta evaluación se hace de manera cualitativa, aunque el servicio posee elementos cuantitativos como el coste o el beneficio de su ejecución.

A modo de resumen, la tabla 2.2 compara las distintas propuestas de formalización analizadas, contrastando entre sí los elementos organizativos que contemplan.

Según la tabla, se comprueba que existen conceptos básicos repetidos en la mayoría de las formalizaciones propuestas, como son los roles o los agentes. En cambio, muchas aproximaciones no realizan un modelado explícito de normas y objetivos, que en algunos casos quedan representados a través de otros conceptos, como las capacidades, que modifican el comportamiento de los agentes, y los objetivos que deben buscar, según las acciones que sean capaces de ejecutar.

En general, todas las formalizaciones analizadas presentan un buen acercamiento para definir una organización de manera formal. Sin embargo, sería interesante que la formalización de la organización mostrara de forma clara cuáles son sus dimensiones (explicadas en la sección 2.1), cuáles son los elementos de estas dimensiones y cómo se relacionan entre sí, tanto de forma estática como dinámica. Por tanto, en la sección 5 presentaremos nues-

Conceptos organizativos								
	OperA	MOISE	PopOrg	LAO	POMF	MACODO	Grossi	Jonker
<b>Dimensión estructural</b>								
<b>Roles</b>	✓	✓	✓	✓	✓	✓	✓	✓
<b>Grupos</b>	✓	✓		✓				
<b>Agentes</b>		✓	✓	✓	✓	✓	✓	✓
<b>Relaciones</b>	✓	✓	✓	✓			✓	✓
<b>Topología</b>				✓				
<b>Dimensión funcional</b>								
<b>Capacidades</b>			✓	✓	✓	✓		
<b>Habilidades</b>			✓	✓				
<b>Servicio</b>			✓		✓			
<b>Objetivos</b>	✓	✓		✓	✓		✓	
<b>Dimensión dinámica</b>								
<b>Interacciones</b>	✓	✓	✓				✓	✓
<b>Dinámica</b>	✓	✓	✓	✓		✓		✓
<b>Dimensión del entorno</b>								
<b>Entorno</b>		✓		✓		✓		✓
<b>Recurso</b>					✓			
<b>Dimensión Normativa</b>								
<b>Normas</b>	✓	✓				✓	✓	
<b>Sintaxis</b>				CTL	L <sub>PR</sub>	Z	Org	TTL
<b>Semántica</b>	LCR			CTL*	T <sub>PR</sub> in L <sub>PR</sub>		Org	

Tabla 2.2: Comparación entre distintas representaciones formales

tra propuesta de modelado de organizaciones, que trata de modelar las organizaciones a partir de sus dimensiones, integrando también los elementos más importantes de las propuestas de formalización vistas en este capítulo. Esta propuesta se basará en trabajos presentados en esta sección [Dig03, DD07, GBKD05, dRCD08].

## 2.4. Artefactos

El modelado del entorno es un reto importante para la comunidad de investigadores de ingeniería del software orientada a agentes (AOSE, por sus siglas en inglés). El entorno se ha convertido en una abstracción de primera clase dentro de un Sistema Multi-Agente, por lo que son muchos los trabajos que se centran en modelarlo [PW07]. Uno de los avances más interesantes en este campo es el framework Agents & Artifacts (A&A) [RVO07]. Esta propuesta compara el mundo de los agentes con los entornos cooperativos humanos, y se caracteriza por tres abstracciones: (i) agentes, los elementos proactivos del sistema, (ii) artefactos, los elementos pasivos del sistema que los agentes usan como apoyo a sus actividades y (iii) workspaces, un contenedor de agentes y artefactos que define la topología del sistema y el espacio donde agentes y artefactos pueden actuar.

Los artefactos [RVO07], por tanto, son entidades reactivas, pero no proac-

tivas, empleadas por los agentes para conseguir sus objetivos. Los artefactos no poseen objetivos asignados, sino que están asociados a los objetivos del agente que usa el artefacto. Para conseguir estos objetivos los artefactos proporcionan una *función*, particionada en varias *operaciones* que los agentes pueden ejecutar al interactuar con el artefacto. Estas operaciones forman parte de la *interfaz de uso* del artefacto, que se completa con las *propiedades observables* que los agentes pueden comprobar sin realizar operaciones en el artefacto. Los artefactos proporcionan un segundo grupo de operaciones, llamadas *operaciones de enlace* (accesibles a través de una interfaz de enlace) que permite la composición de artefactos, además de crear artefactos distribuidos, mediante la unión de artefactos. Estos artefactos podrán estar situados en el mismo workspace o en diferentes workspaces [RVO07] (la porción del entorno que un agente percibe y con la que puede interactuar). Cada workspace contiene un conjunto de artefactos, y el conjunto de workspaces que componen el entorno se usa para definir su topología. Finalmente, los artefactos incorporan una *descripción de su función* (que actúa como un manual) y un conjunto de *instrucciones de uso*, una característica esencial cuando tratamos con sistemas abiertos debido a que agentes externos pueden descubrir artefactos y evaluar si pueden ser útiles para conseguir sus objetivos esperados.

Debido a que los artefactos son componentes del entorno de un SMA que presentan una gran maleabilidad, los diseñadores pueden desarrollar nuevos tipos de artefactos que estén centrados en llevar a cabo diferentes operaciones dependiendo de las necesidades del sistema. Además, es posible utilizar el framework *CArtAgO* [RVO06] para implementar artefactos construidos según los principios del framework A&A [RVO07].

Actualmente, se han desarrollado diferentes tipos de artefactos, los cuales proporcionan una gran variedad de funcionalidades a los desarrolladores de SMA, tales como la coordinación; además de funciones más concretas, como una agenda o un calendario. Existen diversas clasificaciones para los artefactos, siendo la propuesta por Ominici *et al.* en [ORV06] la más extendida. Esta clasificación está centrada en "*quien*" está usando el artefacto y distingue entre *artefactos personales*, que proporcionan funcionalidades a un único agente (p.e. una agenda); *artefactos sociales*, que proporcionan funcionalidades a un grupo de agentes para mejorar la comunicación y la coordinación entre ellos (p.e. una pizarra); y los *artefactos de frontera* (*boundary artifacts*), que envuelve la interacción con sistemas externos y recursos de un SMA (p.e. una base de datos). Esta taxonomía es muy útil para identificar si un artefacto está construido para proporcionar sus funcionalidades a un agente o a un grupo de agentes. Sin embargo, un diseñador de SMA podría necesitar conocer la funcionalidad ofrecida por un artefacto, para poder seleccionar el



más adecuado para ser implementado en su sistema.

En consecuencia, esta sección presenta una nueva taxonomía de artefactos, llamada *taxonomía funcional* [EA10], en la cual los artefactos existentes están agrupados en base a su funcionalidad. Esta taxonomía es lo suficientemente flexible como para incluir nuevos tipos de artefactos que puedan ser desarrollados en el futuro. Esta taxonomía está pensada para ayudar a los desarrolladores de SMA a seleccionar los artefactos más apropiados para mejorar sus sistemas.

### 2.4.1. Taxonomía funcional de artefactos

En esta subsección se propone una nueva taxonomía [EA10] para clasificar artefactos existentes, la cual es también compatible con la clasificación propuesta en [ORV06]. En esta nueva taxonomía, un artefacto puede ser clasificado en una o varias de estas categorías, dependiendo de su funcionalidad: (i) *artefactos básicos*, que comprende artefactos que proporcionan información de características muy generales del mundo (p.e. relojes, calendarios y horarios); (ii) *artefactos de coordinación*, que mejoran la coordinación entre agentes de un SMA; (iii) *artefactos de reputación*, que controlan los valores de reputación que los agentes tienen dentro de una organización; (iv) *artefactos de estigmergia cognitiva*, que proporcionan información sobre un agente o una sociedad de agentes que puede ser útil para otros agentes o grupos de agentes; (v) *artefactos organizativos*, que se usan para controlar una organización; y los (vi) *artefactos de argumentación*, que gestionan las discusiones entre agentes. A continuación, estos tipos de artefactos se explicarán con detalle.

#### Artefacto básico

Los artefactos básicos, o genéricos, proporcionan características generales de uso al sistema. Son capaces de proporcionar información sobre el tiempo, modelando un reloj o un calendario, por ejemplo, como artefactos. También están centrados en proporcionar funciones para ayudar a los agentes (p.e. una calculadora [RP09]). Además, los recursos que pueblan el entorno del sistema pueden ser modelados usando artefactos, facilitando el acceso de los agentes. Por ejemplo, una base de datos puede ser fácilmente modelada usando un artefacto [RP09].

#### Artefacto de coordinación

Este tipo de artefactos se emplea para conseguir la coordinación entre agentes, o mejorar la ya existente. Los artefactos de coordinación [ORV<sup>+</sup>04] encapsulan servicios de coordinación que serán los elementos básicos para

crear entornos efectivos de trabajo colaborativo. Algunos de estos servicios de coordinación son el *Follow Me service* [ORV<sup>+</sup>04] o la *licitación pública* [ROD06]. El servicio de coordinación proporcionado por un artefacto de coordinación puede ser explotado por los agentes en un contexto social. Al ser artefactos, poseen una interfaz y unas instrucciones de uso, pero además proporcionan una especificación de los comportamientos de coordinación, usado por los artefactos para informar formalmente sobre su comportamiento de coordinación, en términos de las reglas de coordinación. El objetivo de los artefactos de coordinación es coordinar acciones entre agentes, dándoles conocimiento sobre sus acciones. Por ejemplo, los agentes puede saber cuándo se ejecuta una acción, el tiempo en que la acción termina o cuál es la siguiente acción a realizar. Finalmente, los artefactos de coordinación establecen guías para los protocolos de acción y comunicación.

Un claro ejemplo de artefacto de coordinación en la vida real podría ser un panel de información colocado en una autovía o autopista, que actúa de manera similar al servicio *Follow Me*. En este caso, cuando un agente de tráfico necesita advertir a los conductores sobre accidentes, inclemencias meteorológicas u otras incidencias en la carretera, publica esta información en los paneles, con lo que los conductores podrán seguir sus recomendaciones. En consecuencia, estos artefactos conseguirán la coordinación que los agentes de tráfico están buscando.

### Artefacto de estigmergia cognitiva

El objetivo de este tipo de artefactos es fomentar el conocimiento dentro de un SMA dando a los agentes información sobre el estado de las entidades del sistema usando la estigmergia [ROV<sup>+</sup>07a]. El conocimiento es un aspecto fundamental para dar soporte a formas emergentes de coordinación, en las que no es necesario definir la coordinación de forma explícita. Los artefactos de estigmergia cognitiva apoyan distintas formas de conocimiento y otras características relacionadas con la estigmergia cognitiva. El conocimiento se fomenta de una de estas formas: (i) de forma personal, informando sobre el estado de un agente concreto, (ii) de forma social, dando información sobre el estado de un grupo de agentes, y (ii) desde el punto de vista del sistema, dando información sobre los eventos que ocurren en el sistema.

Se han definido distintos ejemplos de artefactos para estigmergia cognitiva [ROV<sup>+</sup>07a]: (i) un *tablero* actúa como una interfaz empleada por un agente para representar su intención de usar otro artefacto específico, de interactuar con otro artefacto o de tomar, observar o gestionar anotaciones; (ii) un *registro* realiza un seguimiento de eventos según suceden; (iii) un *diario* permite llevar un seguimiento de las anotaciones hechas por un agente; y (iv)

una *libreta* (*note-board*) se emplea para trazar las acciones individuales de los agentes y las anotaciones. Los mecanismos típicos de estigmergia como difusión, agregación, selección y ordenación [ROV<sup>+</sup>07a] pueden ser integrados en artefactos para apoyar procesos de estigmergia.

Un ejemplo claro de uso de estos artefactos se puede encontrar en las redes sociales como *Facebook*<sup>1</sup>, donde cada usuario tiene su propia página de perfil donde toda su información está almacenada. Modelado como un artefacto, la página de perfil de un usuario llevará a cabo operaciones con artefactos de estigmergia cognitiva. El artefacto *tablero* representaría la interfaz de Facebook, que puede tomar diferentes formas, como la web de Facebook, la app de Facebook para iPhone o la aplicación de Facebook de Xbox Live. El artefacto *registro* representaría los eventos relacionados con un determinado usuario, incluyendo su fecha y hora, tales como '*X e Y son ahora amigos*', '*X se ha unido a un grupo*' o '*X ha escrito en el muro de Y*'. Las anotaciones (actualizaciones del estado de un usuario) hechos de manera intencionada por un usuario se almacenarían en un artefacto *diario*. Por último, el artefacto *note-board* sería una compilación de toda la información almacenada en los artefactos tablero, registro y diario relacionados a un artefacto de página de perfil, además de la información almacenada en las páginas de perfil de otros agentes que pueden ayudar en la estigmergia. Por ejemplo, si un usuario actualiza su estado con una frase como '*Me voy al cine*', algunos de sus amigos podrían coordinarse entre ellos para unirse a ver la misma película. Esta es una característica importante de la estigmergia. La coordinación y organización entre usuarios (agentes) no está impuesta por ellos, pero aparece de forma emergente después de usar la información proporcionada por los artefactos para estigmergia cognitiva.

Además, la existencia de emergencia en un SMA puede producir que ese SMA se convierta en auto-organizativo. En los últimos años, la introducción de propiedades auto-\* dentro de un SMA para convertirlo en auto-organizativo es un campo de investigación clave. Por tanto, los artefactos pueden ser usados para mejorar los SMA con propiedades auto-\* [GVCO08] añadiendo una nueva capa por encima de los recursos del entorno existentes. Para controlar este tipo de artefactos con el objetivo de conseguir las propiedades auto-\* deseadas, se introduce la noción de *agentes del entorno* [GVCO08], que actúan con los artefactos a través de una interfaz de control.

### Artefacto organizativo

Estos artefactos se usan para controlar una organización de agentes, que tienen conciencia de estar dentro de la organización y su propósito es ayudar

---

<sup>1</sup><http://www.facebook.com>

a la organización a alcanzar sus objetivos desde un nivel global y social. Por tanto, los artefactos organizativos están centrados en informar a los agentes de la organización sobre las normas que la gobiernan, su estructura y las acciones que están disponibles dentro de ella.

Un ejemplo que contiene artefactos organizativos es la infraestructura organizativa ORA4MAS [HBKR09]. Los artefactos en ORA4MAS son parte de la infraestructura organizativa, encapsulando funcionalidades referentes al control de la organización. Los artefactos organizativos de ORA4MAS limitan la autonomía de algunas acciones organizativas que los agentes poseen. Además, ORA4MAS presenta cuatro ejemplos de artefactos organizativos: (i) *OrgBoard*, que controla la organización desde una vista global; (ii) *GroupBoard*, que controla el ciclo de vida de un grupo de agentes; (iii) *SchemeBoard*, que apoya y controla la ejecución de un esquema social; y (iv) *NormativeBoard*, que almacena la información sobre las normas.

En [HBO10], los autores presentan otro ejemplo de artefacto organizativo, el *artefacto de coordinación de evolución de roles*, que puede lidiar correctamente con conceptos organizativos. El objetivo de este artefacto es construir y evolucionar una taxonomía de especialización de roles, que consiste en un conjunto de roles con un orden temporal concreto; haciendo que esta información esté disponible para los agentes.

## Artefacto de reputación

Estos artefactos se usan para establecer una evaluación para cada agente, que se usa por el resto de agentes que pueblan el sistema para construir su propia reputación sobre un agente.

Por ejemplo, el artefacto de reputación diseñado en ORA4MAS trabaja de esta manera [HBV08]: los artefactos *GroupBoard* y *SchemeBoard* de ORA4MAS recolectan todas las normas violadas y las envían al artefacto de reputación, que es responsable de calcular una evaluación para los agentes dentro de la organización, que será presentada como una propiedad observable del artefacto. La evaluación proporcionada por este artefacto de reputación es un instrumento para influenciar la reputación de un agente.

Los artefactos de reputación pueden ser usados, por ejemplo, para calcular puntuaciones para un congreso, como el *Impacto Estimado de una Conferencia* (EIC), como hace el *Computer Science Conference Ranking*<sup>2</sup>. Después de que el artefacto asigne una puntuación al congreso (es decir, realice una evaluación), la comunidad de investigadores (es decir, agentes) estará influenciada por esta puntuación y su conocimiento previo del congreso, con lo que construirá su propio valor de reputación sobre este congreso.

<sup>2</sup><http://www.cs-conference-ranking.org>

## Artefacto de argumentación

Se emplea para evaluar la aceptación social de argumentos para ayudar en el proceso de toma de decisiones entre un grupo de agentes sobre acciones que deben ejecutar. El objetivo principal de este tipo de artefactos es el de proporcionar coordinación en el proceso de argumentación que los agentes deben llevar a cabo para alcanzar un acuerdo. Este artefacto recolecta los argumentos dados por los agentes participantes en el proceso de argumentación y los evalúa para proporcionar dos tipos de valores de aceptación (aceptabilidad social y comportamiento social), así como proponer un acuerdo factible para la sociedad de agentes. La 'aceptabilidad social' se refiere a la admisibilidad de los argumentos que un agente produce dentro de la sociedad. El 'comportamiento social' proporciona una vista global del comportamiento de la sociedad de agentes en el sistema. Ambos tipos de aceptabilidad pueden ser consultados por los agentes a su discreción.

En consecuencia, este artefacto no sólo ayuda a la coordinación, sino que además ayuda en el proceso de argumentación y en las tecnologías del acuerdo [LM08]. Un ejemplo claro de este tipo de artefactos es el *Artefacto de Co-Argumentación* (CAA, por sus siglas en inglés) [OMO08], que se usa para controlar los argumentos y también encapsula un servicio de coordinación. El CAA actúa como un mediador de la interacción entre los agentes y ayuda a una implementación simplificada del sistema de argumentación de un SMA.

Otro ejemplo de artefacto de argumentación puede encontrarse en un programa de televisión interactiva donde se está llevando a cabo una discusión con dos posibles respuestas ('sí' o 'no'). Se pide a los espectadores que envíen un mensaje SMS con su respuesta ('sí' o 'no') seguido de un argumento que la apoya. Estos argumentos se envían al artefacto de argumentación que los evalúa y proporciona una respuesta, que no está sólo centrada en los porcentajes, sino que también lo está en los argumentos que apoyan estas respuestas de 'sí' o 'no'. Además, los espectadores son capaces de conocer la aceptación que sus argumentos han tenido en el resto de audiencia del programa.

### 2.4.2. Relaciones entre categorías

Las operaciones de enlace que existen en los artefactos permiten a éstos usar otros artefactos, por lo que podrían aparecer relaciones entre artefactos pertenecientes a diferentes categorías de nuestra taxonomía, tal y como vemos en la Fig. 2.3.

Por ejemplo, uno de los objetivos del proceso de argumentación es alcanzar un acuerdo entre agentes para conseguir coordinarlos. Por tanto, podría ser necesario usar un artefacto de coordinación cuando se usa un artefacto de

Ejemplos de artefactos				
Nombre del artefacto	Clase	Prop. observables	Operaciones	Op. de enlace
Artefactos básicos				
Agenda [RP09]	Personal	next_todo, last_todo	setTodo, cancelTodo	-
Counter [RP09]	Social	count	inc, reset	-
Flag [RP09]	Social	state	switch	-
Database [RP09]	Frontera	n_records, table_names	createTable, addRecord, query	-
Artefactos de coordinación				
Follow Me [ORV <sup>+</sup> 04]	Social	act(a): se ha hecho a	do(a): ejecutar acción a get: siguiente acción a ejecutar	-
Pub. Comp. Tender [ROD06]	Social	Sin especificar	Sin especificar	Sin especificar
Artefactos de argumentación				
Argument Acceptance Artifact [OMO08]	Social	Social Acceptability Social Behavior	Write Arguments	-
Artefactos de estigmergia cognitiva				
Dashboard [ROV <sup>+</sup> 07a]	Social	Annotations	takeAnnotations, manageAnnotations	traceOperations
Log [ROV <sup>+</sup> 07a]	Social	Events	inspectEvents, orderEvents	traceOperations
Diary [ROV <sup>+</sup> 07a]	Personal	Annotations	makeAnnotation	-
Note-board [ROV <sup>+</sup> 07a]	Social	Annotations	makeAgentAnnotation	makeArtifactAnnotation
Artefactos organizativos				
OrgBoard [HBKR09]	Social	OrgSpecification Group, Scheme and Normative Boards	getOrgAgents getMemberAgents	registerOrgArt
GroupBoard [HBKR09]	Social	OrgBoard, SchemeBoards Type, PlayableRoles PlayersOfRole	adoptRole leaveRole	addScheme removeScheme isMember
SchemeBoard [HBKR09]	Social	OrgBoard, GoalsState NormativeBoard ResponsibleGroupBoards Type, PlayableMissions PlayersOfMissions	commitMission leaveMission setGoalAchieved	-
NormativeBoard [HBKR09]	Social	OrgBoard, NormStatus	-	updateAgentStatus
Role Evolution Coordination Artifact [HBO10]	Social	Taxonomy	getBestRolesForInteraction getAgentsForRoles, getRolesForAgent communicateTrust	-
Artefacto de reputación				
Alice's MSc course [HBV08]	Social	Evaluation	-	informAboutOrganization

Tabla 2.3: Ejemplos de artefactos agrupados por su tipo

argumentación. De forma similar, los artefactos organizativos pueden explorar los artefactos de coordinación para alcanzar los objetivos organizacionales del sistema. Los artefactos de coordinación pueden ser empleados también por los artefactos de reputación, debido que los agentes pueden estar interesados en usar un servicio de coordinación encapsulado dentro de un artefacto de coordinación, y recomendado por un agente, si el agente recomendador tiene una buena reputación. Para calcular esta reputación, el agente que recibió la recomendación puede solicitar al artefacto de reputación una evaluación del agente recomendador.

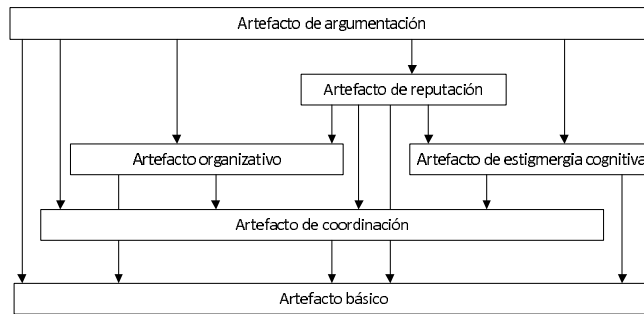


Figura 2.3: *Relaciones entre las categorías de la taxonomía funcional de artefactos*

Los artefactos organizativos son también útiles para otras categorías. De esta forma, los artefactos de reputación pueden necesitar artefactos organizativos para seguir los eventos que ocurren en una organización. Además, los artefactos organizativos pueden ayudar a los artefactos de argumentación a alcanzar un acuerdo entre agentes proporcionándoles información sobre aspectos organizativos.

Finalmente, los artefactos de reputación pueden ser utilizados por los artefactos de argumentación, ya que un argumento proporcionado por un agente puede ser más fiable si ese agente tiene una buena reputación. Además, si es necesario mejorar el conocimiento de un artefacto de reputación, puede hacer uso de un artefacto de estigmergia cognitiva.

### 2.4.3. Discusión

Presentados originalmente como uno de los dos conceptos principales del framework Agents & Artifacts (A&A), los artefactos son entidades reactivas, pero no proactivas, que ayudan a los agentes a alcanzar sus objetivos. Los artefactos se pueden clasificar de diversas formas, pero actualmente la taxonomía más extendida clasifica los artefactos según quien hace uso de ellos. Sin embargo, esta clasificación no proporciona información del propósito que tienen los artefactos, es decir, para qué están hechos.

Por tanto, en esta sección hemos presentado una nueva taxonomía centrada en el propósito y la funcionalidad de los artefactos. Esta taxonomía ayudará a los desarrolladores de SMA a conocer las funciones que los artefactos son capaces de hacer, para que puedan seleccionar los artefactos que mejor se ajusten a las necesidades de su sistema.

A modo de resumen, la tabla 2.3 muestra diferentes ejemplos de artefactos para cada una de las categorías definidas en nuestra taxonomía. Por cada artefacto se indica la clase a la que pertenece, según la taxonomía de [ORV06],

sus propiedades observables, sus operaciones y sus operaciones de enlace.

La taxonomía funcional propuesta tiene en cuenta todos los artefactos desarrollados hasta ahora. Sin embargo, ya que los artefactos son todavía un enfoque en desarrollo, podrían aparecer nuevos ejemplos en el futuro, que pueden proporcionar importantes funcionalidades que no han sido consideradas hasta el momento. Estas funcionalidades pueden suponer nuevas categorías dentro de nuestra taxonomía, enriqueciéndola.

## 2.5. Mecanismos organizativos

Esta sección describe el concepto de mecanismos organizativos [CBHO09], una propuesta centrada en mejorar la coordinación entre los agentes de un sistema multiagente.

Los mecanismos organizativos [CBHO09] pueden ser un método válido para proporcionar coordinación dentro de las organizaciones. Son mecanismos introducidos en un SMA con el objetivo de influir en el comportamiento de los agentes hacia una mayor efectividad con respecto al propósito global del sistema, asumiendo que los agentes que participan en el sistema son racionales. Es decir, intentan maximizar su utilidad con cualquier acción que lleven a cabo. Por lo tanto, estos mecanismos puede proporcionar información adicional a los agentes que pueden persuadirlos para comportarse de una forma determinada; o pueden producir cambios en el entorno que pueden imponer ciertos comportamientos a los agentes. Así, es muy útil usar estos mecanismos en un sistema abierto donde se localizan los agentes externos, siendo capaces de fomentar la coordinación.

Existen dos tipos de mecanismos organizativos: informativos y regulativos.

Los **mecanismos organizativos informativos** [CBHO09] se definen como una función que dada una descripción del estado interno de un agente, y teniendo en cuenta la descripción parcial que el mecanismo tiene del estado actual del entorno, proporciona información, que puede consistir en un conjunto de acciones que un agente puede realizar pero posiblemente no tenga conocimiento de ellas, una recomendación de una acción que eventualmente puede ser una 'buena acción' para el agente, o información sobre las consecuencias que una acción dada puede tener.

Todos los mecanismos informativos tienen en común que su uso no es impuesto. Los agentes son libres de usar estos mecanismos a su discreción. Para usar un mecanismo informativo un agente debe proporcionarle parte de su estado interno. De hecho, cuando se asume racionalidad en los agentes [RG97], éstos deben usar un mecanismo organizativo informativo dado



si y sólo si esperan que el uso de este mecanismo sea ventajoso para ellos. Los mecanismos organizativos informativos pueden mejorar el rendimiento de agentes individuales y pueden tener efectos en el rendimiento global de un SMA organizado con respecto a una función global de utilidad. La información proporcionada por este tipo de mecanismos mejorará el conocimiento de un agente, debido a que esta última incluye información adicional para el razonamiento y por tanto mejorará su elección de futuras acciones.

Los **mecanismos organizativos regulativos** [CBHO09] comparten el mismo objetivo que los mecanismos informativos, pero están centrados en introducir cambios en el entorno para mantener a los agentes alejados de comportamientos indeseables que llevarán al sistema a estados no productivos, es decir, estos mecanismos están al cargo de producir cambios en el sistema para alcanzar estados que mejoren la utilidad global del sistema. La razón detrás de introducir cambios en el entorno es que los agentes perciben estos cambios, por lo que posiblemente alteran su razonamiento para decidir la siguiente acción a ejecutar. Este tipo de mecanismos confían en la existencia de un diseñador del sistema, que define la relación de preferencia sobre estados del sistema representados a través de la función global de utilidad, que tiene la autoridad suficiente para imponer ciertos cambios en el sistema.

Se consideran dos tipos de cambios posibles en el entorno: (i) introducción de incentivos para hacer que los agentes sigan un comportamiento deseado y (ii) cambios en el espacio de acciones de los agentes. En consecuencia, se han definidos dos tipos de mecanismos regulativos [CBHO09]:

- Un **mecanismo incentivo** es una función que produce cambios en la distribución de probabilidad de transición de un SMA dada una descripción parcial del estado del entorno de un SMA.
- Un **mecanismo coactivo** es una función que produce cambios en la función de capacidad de los agentes, añadiendo o eliminando acciones desde el espacio de acciones de los agentes, dada una descripción posiblemente parcial de estado del entorno del SMA.

Los mecanismos incentivos pueden producir cambios en las consecuencias de las acciones de los agentes introduciendo recompensas y castigos. Obviamente, las recompensas y los castigos pueden producir variaciones en la utilidad esperada de las acciones de un agente y, por lo tanto, los agentes racionales cambiarían sus decisiones en consecuencia (si tienen conocimiento de estos incentivos). Por tanto, los agentes deben estar informados sobre las normas que rigen el sistema. En el caso de mecanismos coercitivos los cambios en el sistema son producidos en el espacio de acciones de los agentes. Nuevas acciones pueden ser añadidas o acciones existentes pueden ser eliminadas.

Ambos tipos de mecanismos emergen como una importante contribución a los SMA. Debido que hoy en día los SMA son abiertos y heterogéneos, con lo que es posible que agentes no colaborativos pueblen un sistema, es necesario estar provistos de mecanismos que ayuden a los administradores del sistema a mantener el SMA bajo control. Los mecanismos informativos y regulativos son muy útiles para afrontar esta tarea. Aparte de esto, los mecanismos organizativos necesitan estar implementados en el entorno de un SMA. La información proporcionada por los mecanismos organizativos puede ser proporcionada a los agentes usando artefactos, siendo una contribución para mejorar el entorno de los SMA.

En el capítulo 3 presentaremos nuestra propuesta sobre cómo modelar los mecanismos organizativos a través de artefactos.

## 2.6. Adaptación de Sistemas Multiagente Centrados en Organizaciones

Esta sección presenta un estado del arte sobre conceptos de adaptación en sistemas multiagente centrados en las organizaciones, mostrando distintas técnicas empleadas para introducir de forma dinámica modificaciones tanto en los agentes que componen la organización como en elementos estructurales como la topología o las normas que rigen el sistema.

### 2.6.1. Concepto de adaptación

Según la RAE<sup>3</sup>, adaptación significa 'Acción y efecto de adaptar o adaptarse'. La palabra adaptar tiene distintos significados, como 'Acomodar, ajustar algo a otra cosa', 'Hacer que un objeto o mecanismo desempeñe funciones distintas de aquellas para las que fue construido' o 'Dicho de un ser vivo: Acomodarse a las condiciones de su entorno'. Pese a no ser un ser vivo, la última definición es la que mejor describe el objetivo que perseguimos al usar adaptación dentro de un SMA. Por tanto, y pese a que la definición de adaptación dentro del área de SMA no está consensuada (al igual que pasa con otras definiciones como la de acuerdo o contrato), entendemos que un SMA es adaptativo cuando es capaz de adaptarse a los cambios que se producen en las condiciones, requisitos o el entorno de dicho sistema [PHBG09].

En el ámbito de la Teoría de Organizaciones [GS98], este concepto se define de varias formas, desde una elección estratégica, referida a la conse-

---

<sup>3</sup>Diccionario de la Lengua Española, Real Academia Española, Vigésima segunda edición

cución de los fines previstos sobre la base de una evaluación racional de los medios disponibles y las condiciones, resultando en una decisión explícita de cambiar la organización; a un determinismo del entorno, en el que el cambio organizacional es una respuesta a los requisitos del entorno.

En el ámbito de los sistemas multiagente el término adaptación no tiene una definición uniforme. Dignum [Dig09] considera la adaptación como un factor del diseño que requiere de una decisión, que da como resultado la modificación de algunas de las características de la organización. Estas decisiones pueden venir de dos vertientes distintas: la temporalidad y la intencionalidad. Así, de una parte, existen dos formas de cambios de tipo temporal [DSD04]. Un cambio temporal es *proactivo* cuando la modificación prepara al sistema para los cambios impredecibles que puedan producirse y *reactivo* si el cambio en la organización se realiza tras la ocurrencia de un evento dado. De otra parte, los cambios se pueden clasificar según su intencionalidad. En este caso el cambio puede ser *ofensivo* si lo que se busca es una ventaja a la hora de competir, o *defensivo*, si se pretende la supervivencia de la organización.

Para hacer una reorganización proactiva los agentes deben estar equipados con mecanismos para razonar y evaluar el comportamiento actual y el deseado y su utilidad. En el caso de reorganización reactiva, los agentes deben ser capaces de sentir y reaccionar a cambios del entorno, pudiendo ser agentes más simples.

Se pueden combinar estos factores para obtener las cuatro maniobras de cambio posibles [EAP97]:

- **Preventivo** (proactivo, ofensivo): toma ventaja de posibles eventos futuros y resulta útil cuando el futuro es impredecible y cuando la innovación es un elemento de competición. Por ejemplo, un sistema de gestión de una universidad podría introducir nuevos agentes con el rol profesor en previsión de un aumento de alumnos para el siguiente curso, aún sin estar seguro de este incremento.
- **Protectivo** (proactivo, defensivo): aplicado para limitar los daños causados en un futuro. Este tipo de cambio se produce, por ejemplo, cuando se incrementa la seguridad en un lugar determinado por la visita de autoridades, teniendo que añadir nuevos agentes y cambiando la organización de la seguridad que habitualmente presenta ese lugar.
- **Explotador** (reactivo, ofensivo): ocurre después de un evento, para explotar las oportunidades que hayan surgido de su ejecución. Por ejemplo, después de la celebración de un gran evento, como la America's Cup en el puerto de Valencia, estas instalaciones han sido posteriormente utilizadas como zona de ocio y parte del circuito de Fórmula 1,

teniendo que adaptar aspectos estructurales de la ciudad como la red de autobuses y tranvía para facilitar el acceso de los ciudadanos a esa zona.

- **Correctivo** (reactivo, defensivo): usado para prevenir más daño, cuando fallan otras tácticas, y asegurar así la existencia. En un entorno empresarial, si una compañía está perdiendo mucho dinero, reorganizará su estructura eliminando departamentos y despidiendo empleados para minimizar las pérdidas.

Cernuzzi y Zambonelli [CZ06, CZ10] entienden la adaptación como la facilidad que tiene una metodología de desarrollo de sistemas multiagente para adaptarse a los cambios producidos en el sistema durante la fase de diseño. En nuestro caso, debemos centrarnos en los cambios producidos durante la ejecución del sistema, en la que serán los propios agentes los que se hagan cargo de estos cambios.

### A. Tipos de cambio

A la hora de realizar un proceso de adaptación en un SMA orientado a la organización, podemos distinguir dos tipos de cambio: de comportamiento y estructurales [Dig09].

Los cambios de comportamiento son aquellos en los que la estructura del sistema se mantiene, mientras los agentes cambian su protocolo de comportamiento. Existen tres tipos de cambios en el comportamiento:

- *Un nuevo agente entra en el sistema.* Es necesario un acuerdo para entrar en la organización, tomando un rol determinado, indicando derechos y deberes del agente una vez ha tomado ese rol.
- *Un agente deja el sistema.* Es necesario determinar si la operación es posible, teniendo en cuenta determinadas condiciones impuestas por el SMA. En ocasiones, puede no resultar adecuado que un agente con un determinado rol abandone el sistema. En otros momentos, será conveniente reasignar ese rol tan rápido como el agente que lo posee abandone el sistema.
- *Instanciación del patrón de interacción.* Un cambio de este tipo consiste en que dos agentes llevando a cabo un determinado patrón de interacción llegan a un acuerdo para seguir un protocolo que se ajuste a este patrón de interacción.

El segundo tipo de cambio es el llamado *cambio estructural*. En este caso, se producen modificaciones en elementos estructurales del sistema. Como se ha definido en la sección 2.6.1 existen dos formas de realizar un cambio en la organización. Por tanto, los cambios estructurales podrán ser de dos tipos:

- Auto-diseño organizativo: implica la emergencia de los cambios, aparecidos por la interacción entre agentes a nivel local, que generan los cambios a nivel global de la organización.
- Adaptación estructural: las sociedades diseñadas se adaptan a los cambios en el entorno añadiendo, borrando o modificando sus elementos estructurales (roles, dependencias, normas, ontologías o primitivas de comunicación).

Para referirse a los cambios de auto-diseño organizativo el término auto-organización (*self-organization*) es el más extendido en el estado del arte. Estos cambios son de tipo *bottom up* (de abajo hacia arriba) y los SMA que implementan técnicas de auto-organización son también conocidos como *sistemas adaptativos* [CZ10], donde un cambio en los comportamientos individuales de los agentes dará lugar a un cambio en la organización de forma emergente. Entendemos la *self-organization* como un proceso endógeno (realizado por los propios agentes) y *bottom-up* en los sistemas en los cuales los agentes, que no están al tanto de la existencia de la organización como un todo, sólo manipulan la información a nivel local para adaptar el sistema a las presiones del entorno modificando indirectamente la organización. Por tanto, los agentes cambian, por interacciones locales y propagación, la configuración (topología, vecinos, influencias, diferenciación) o el entorno del sistema. En esta aproximación existen numerosos trabajos realizados, una muestra de los cuales se repasará en la sección 2.6.2.

El segundo enfoque, la adaptación estructural, es más conocida como *re-organización*. Los sistemas que siguen esta aproximación son conocidos también como *sistemas adaptables*, siguiendo un enfoque *top-down* (de arriba a abajo) para SMA basados en organizaciones, donde un cambio en algún aspecto organizativo producirá cambios en los agentes que componen la organización. La reorganización es un proceso endógeno (controlado por los propios agentes del sistema) o exógeno (controlado por el usuario o por un sistema externo), referido a sistemas en los cuales la organización se manipula explícitamente mediante especificaciones, restricciones u otros medios, para asegurar un comportamiento global adecuado cuando la organización no es apropiada. Los agentes se encuentran al tanto del estado de la organización y su estructura, y son capaces de manipular primitivas para modificar

su entorno social. Este proceso puede ser iniciado por una entidad externa o por los propios agentes, razonando directamente sobre la organización (roles, especificación organizacional) y los patrones de cooperación (dependencias, compromisos, poderes). Esta forma de adaptación no está tan estudiada como la auto-organización, pero se dará una muestra de los trabajos en esta materia en la sección 2.6.3.

## 2.6.2. Self-organization

Existen numerosas definiciones y aproximaciones al concepto de auto-organización (*self-organization*). Esta sección se centrará en repasar algunas de las aportaciones más interesantes, para conseguir una visión general en el campo de los sistemas multiagente auto-adaptativos.

En la sección 2.6.1 se expuso que el concepto de adaptación no está consensuado entre la comunidad de investigadores en sistemas multiagente. En el caso de los sistemas auto-organizativos tampoco existe una definición clara sobre estos sistemas, aunque diversos autores han aportado las características que deben cumplirse en un sistema para ser considerado como auto-organizativo.

Por una parte, De Wolf *et al.* [DWH05] establecen que se debe realizar un buen análisis de requisitos antes de decidir si usar un sistema auto-organizativo. Según los autores, estos sistemas deberían ser capaces de mantener ciertas propiedades a nivel macroscópico durante toda la ejecución del sistema, además de permitir que estas propiedades sean medibles para poder realizar una evaluación del comportamiento del sistema. También establecen que un SMA auto-organizativo debe cumplir las siguientes propiedades:

- La información disponible se encuentra *descentralizada* o distribuída, con lo que sólo se producirán interacciones locales para conseguir esta información.
- El sistema debe ser capaz de lidiar con la componente dinámica, siendo necesario que sea *robusto*, para mantener el comportamiento macroscópico sin cambios, y poseyendo *flexibilidad* y *adaptabilidad* para poder llevar a cabo las reconfiguraciones locales que se producirán en el sistema.

La robustez se prefiere por encima de la optimalidad, ya que la segunda sólo podrá ser conseguida cuando el sistema permanece estático. De Wolf también describe que los sistemas multiagente emergentes auto-organizativos prometen ser escalables, robustos, estables, eficientes y de baja latencia [Ant04],

pero se comportan de manera no determinista, aunque poseen tendencias que son predecibles después de comprobar la evolución del comportamiento macroscópico tras una serie de ejecuciones del sistema.

Por otra parte, Calta [Cal09] establece que los sistemas auto-organizativos deben tener las siguientes características:

- **Estructura.** Cuando un sistema está organizado debe poseer una estructura. Teniendo una estructura se restringe el grado de libertad de los componentes del sistema, minimizando las posibilidades de obtener comportamientos no deseados en el sistema.
- **Adaptabilidad.** Si la estructuración del sistema se ve afectada por cualquier suceso, el sistema debe ser capaz de adaptarse a una nueva estructura.
- **Descentralización.** No hay un elemento centralizado controlando la auto-organización, por lo que los componentes son autónomos.
- **Interacción y conocimiento local.** Los componentes no tienen una vista global del sistema. Cada componente sólo tiene visión de sí mismo y de su entorno más próximo.
- **Homogeneidad.** Los elementos del sistema tienen las mismas habilidades. Si existe un elemento con habilidades diferentes (por ejemplo, un líder), estas habilidades superiores habrán aparecido a causa de la auto-organización del sistema.
- **Comportamiento no determinista del sistema.** El objetivo de un sistema auto-organizativo es conseguir un comportamiento deseado a nivel global. Los pasos seguidos para alcanzar este comportamiento no son conocidos de antemano, y se podrían dar distintas combinaciones de acciones que alcancen el estado global deseado.

Tanto De Wolf como Calta coinciden en los aspectos de adaptabilidad, descentralización, localidad y no determinismo, pudiéndose considerar éstas como las propiedades que un SMA debe mostrar para ser considerado auto-organizativo.

En su trabajo, Kota [KGJ09] presenta otra definición de auto-organización: 'mecanismo o proceso que permite al sistema cambiar su organización sin necesidad de una actuación externa y explícita'. La auto-organización puede consistir en formar una organización a partir de entidades desordenadas o de reordenar una organización existente. Presenta además un marco de trabajo consistente en un modelo de organización simple, en el que el proceso de

adaptación se lleva a cabo por todos los agentes para mejorar la actuación del sistema. Las modificaciones se llevan a cabo a nivel inter-agente, modificando interacciones entre agentes, y no acarrea ninguna modificación en el comportamiento interno de los agentes.

Se han presentado diversos trabajos para diseñar este tipo de sistemas. La metodología más representativa para el desarrollo de sistemas multiagente auto-organizativos es ADELFE [BGPP03]. Esta metodología se basa en los principios de la teoría AMAS (*Adaptive Multiagent Systems*) [GCG99], que propone el diseño de agentes que posean las habilidades de modificar de manera autónoma y local sus interacciones para reaccionar a los cambios que se produzcan en el entorno. Cada agente tiene sus propios objetivos dentro del sistema y cooperará con otros agentes para alcanzarlos, intentando evitar las Situaciones No Cooperativas (*Non Cooperative Situations*, NCS) como conflictos o concurrencia, adaptándose cuando sea necesario para llegar a una situación de cooperación entre agentes.

La auto-organización puede ser conseguida mediante distintas herramientas. En la sección 2.4 de este trabajo se han presentado los artefactos, una entidad introducida en el entorno de un sistema multiagente que los agentes utilizan para alcanzar sus objetivos. En [GVCO08], Gardelli *et al.* usan los artefactos como una herramienta para introducir auto-organización dentro de un sistema multiagente. Para alcanzar estas propiedades se introduce en el sistema el concepto de agentes del entorno, encargados de controlar los artefactos. El uso de estos artefactos se realizará a través de una interfaz de gestión, habitualmente cerrada para los agentes de usuario que pueblan el sistema. Estos agentes de usuario no son capaces de percibir la existencia de los agentes del entorno, pero sí los artefactos, a los que también tendrán acceso.

Su aproximación está basada en la simulación y sigue las siguientes fases:

- **Modelado.** Se desarrolla un modelo abstracto del sistema, caracterizando a agentes de usuario, artefactos y agentes del entorno.
- **Simulación.** Se realizan simulaciones del comportamiento del sistema para encontrar los parámetros que serán interesantes para el ajuste del modelo.
- **Ajuste.** En esta última fase, tanto los agentes del entorno como los parámetros escogidos en la fase de simulación se modifican hasta que se consigue la dinámica del sistema deseada.

Uno de los conceptos de los que más se nutren los estudios en auto-organización es el de *estigmergia* [Gra59]. Este concepto fue introducido por



Glassé a finales de los años 50 mientras intentaba estudiar el comportamiento de los insectos sociales y se definió como una clase de mecanismos que median en las interacciones entre animales, lo que es fundamental para conseguir formas emergentes de comportamiento coordinado a nivel social. Originalmente, el concepto de estigmergia se utilizó para explicar la llamada *paradoja de la coordinación* entre los niveles social e individual: esto es, aunque un grupo de insectos parezca comportarse de forma organizada y coordinada a nivel global, cuando nos fijamos en el comportamiento de un único individuo comprobamos que se comporta como si estuviera solo, sin interactuar con otros insectos ni estar envuelto en ningún comportamiento colectivo. La explicación a este fenómeno es que los insectos interactúan de forma indirecta, usando por ejemplo feromonas en el caso de las hormigas para construir el recorrido que se debe seguir. Por tanto, los insectos cambian su comportamiento según el entorno, y para realizar cambios en él que lleven a un cambio global, los insectos cuentan con herramientas, como las feromonas.

En el campo de la informática en general, y en el de los SMA en particular, se ha usado la estigmergia como una técnica para resolver problemas complejos, pero en los últimos años se está convirtiendo en un acercamiento para diseñar y desarrollar sistemas. En general, el uso de estigmergia en los SMA se reduce a intentar imitar el comportamiento de las hormigas, sin tener en cuenta que los agentes son entidades cognitivas (al contrario que las hormigas) y que los entornos en los que se mueven son más complejos y poseen una funcionalidad mayor que la de ser un simple contenedor de feromonas.

Por este motivo, diversos investigadores han comenzado a introducir elementos cognitivos cuando estudian la estigmergia, estudiando por ejemplo la estigmergia en actividades humanas [VDP06]. Un trabajo interesante en la materia es el de Ricci *et al.* [ROV<sup>+</sup>07b], en el que definen la estigmergia cognitiva. En su trabajo, se consideran agentes con capacidades cognitivas, que pueden ser racionales, heterogéneos y con capacidad de adaptación y aprendizaje. Uno de los elementos importantes de este trabajo es el *entorno*, que gana complejidad y en el que los agentes conocen (i) su campo de trabajo, (ii) que comparten el entorno con otros agentes, y (iii) las funcionalidades y oportunidades del entorno que pueden explotar para alcanzar sus objetivos. Por tanto, el entorno deja de ser visto como un contenedor para pasar a ser una entidad de primer orden que contiene herramientas, que pueden ser utilizadas tanto a nivel individual como a nivel social. Estas herramientas son los *artefactos*, explicados con detalle en el capítulo 2 (sección 2.4).

Finalmente, cabe destacar que la comunidad de investigadores en auto-organización se reúne en torno a conferencias como SASO (*IEEE Inter-*

*national Conference on Self-Adaptive and Self-Organizing Systems*)<sup>4</sup>. Esta conferencia se define como un foro para sentar las bases de un nuevo enfoque a la ingeniería de sistemas, redes y servicios basados en la auto-adaptación y auto-organización. A tal fin, el encuentro tiene como objetivo atraer a participantes de distintos orígenes, para fomentar la interacción entre diferentes campos de investigación, y para exponer y discutir las teorías innovadoras, marcos, metodologías, herramientas y aplicaciones.

### 2.6.3. Reorganización

La sección anterior ha dado una visión de los sistemas en los que se produce auto-organización. Desde esta sección nos centraremos en la reorganización, campo en el existen trabajos interesantes como el de Dignum y Dignum [Dig09] o el de Picard *et al.* [PHBG09].

De los cambios estructurales será interesante contemplar los cambios de adaptación estructural. Más en detalle, este tipo incluirá cambios en la especificación de la organización, la forma en la que la organización se construye y en la asignación de roles. Por tanto, debemos contemplar cambios en la forma en la que se estructuran las unidades organizativas, así como cambios en roles, normas, objetivos organizacionales y otros elementos propios de las organizaciones virtuales. El añadido, borrado y modificación de alguno de estos elementos propiciará un cambio en la organización al que los agentes deberán adaptarse.

Según Picard *et al.* [PHBG09] se deben contemplar cambios en la propia definición de la organización y en la asignación de roles a agentes, así como en la forma en la que la organización multiagente se construye. La reorganización puede consistir en la generación y composición de agentes en el sistema para reasignar los roles dados. La razón del cambio sería conseguir que la estructura se adapte al entorno y a las tareas que deben completar los agentes. Según Glaser [GM97] la organización se cambia con la entrada de un nuevo agente jugando un rol particular y sólo es aceptada su entrada si mejora la función de utilidad del sistema.

#### A. Realización del cambio

Como hemos comentado en la sección 2.6.1, existen dos tipos de cambio si nos referimos a criterios temporales [DSD04]. El cambio puede comenzar a causa de un criterio previamente definido, por lo tanto es un cambio *proactivo*. Estos cambios intentan mejorar el comportamiento global de la organización, adelantándose además a futuros contratiempos que puedan surgir

---

<sup>4</sup>[www.saso-conference.org](http://www.saso-conference.org)

en el entorno de la organización. Si la adaptación se produce tras un cambio en el funcionamiento del sistema, se produce un cambio de tipo *reactivo*. En general, el cambio es necesario si no se consiguen determinados objetivos, estando sujeto a la utilidad de la organización en general y a la del agente en particular. Por supuesto, estos cambios se producen en tiempo de ejecución.

Por su parte, siguiendo con criterios temporales, Picard *et al.* [PHBG09] definen que la decisión sobre cuándo empezar el proceso puede ser *estática* o *dinámica*. En el primer caso, el proceso comienza a causa de un criterio definido de antemano. En el segundo caso, se dispara el cambio a consecuencia del funcionamiento del sistema. Si los agentes no consiguen determinados objetivos, la organización puede cambiar. El momento del cambio está sujeto a la utilidad de la organización y a la del agente.

El encargado de ordenar un cambio en el sistema podrá ser un agente externo [CZ10] o un actor del mismo [GJW08]. Este actor podrá ser un agente, o un grupo de agentes del sistema, o por el contrario un humano. Los actores encargados del cambio deberían tener un rol especial dentro de la organización, por ejemplo llamado *system adapter* que les denote como encargados de realizar modificaciones dentro de los elementos estructurales de la organización.

En la sección 2.6.1 se ha especificado que la adaptación de un SMA aparece cuando se dan cambios en las condiciones, requisitos y entorno del Sistema Multiagente [PHBG09]. Como requisitos podemos entender los requisitos de los grupos de interés participantes en el sistema [APDD08]. A ellos se dirige la organización y por ellos se puede producir un proceso de adaptación que ayude a responder a sus necesidades. Las condiciones y requisitos del sistema se ven reflejados en los objetivos del sistema. El fin de un sistema multiagente centrado en la organización es conseguir sus objetivos globales. Si se detecta que la organización no está llegando a la consecución de estos objetivos, la organización deberá cambiar alguno de sus parámetros, como por ejemplo la topología, las normas o los roles de la misma.

El segundo aspecto por el que decidir si es necesaria una reorganización son los cambios en el entorno del sistema [APDD08]. Es importante tener en cuenta los sucesos que ocurren en el entorno, ya que repercutirán dentro de nuestra organización. Es por eso que se deben controlar el resto de organizaciones que pueblan el entorno. Un cambio en alguno de sus aspectos podría repercutir en las condiciones de nuestra organización. Por ejemplo, otras organizaciones podrían decidir enviar un gran número de agentes a nuestra organización o podrían decidir llegar a un acuerdo con nuestra organización para conseguir un determinado objetivo. También es necesario comprobar el comportamiento de los agentes que pueblan el entorno (sin estar asociados a ninguna organización) para ver si deciden o no entrar en nuestra organiza-

ción, y con ello podríamos necesitar de una reorganización.

Otros elementos del entorno, como los recursos y servicios disponibles, también jugarán un papel clave a la hora de decidir sobre la necesidad de adaptar nuestra organización. Si un recurso o servicio que se proporcionaba en el entorno deja de estar disponible, es posible que el sistema necesite de cambios para poder afrontar esa pérdida. También necesitaríamos de una reorganización si desde el entorno se proporcionaran recursos y/o servicios que cubriesen necesidades del sistema o si dichos recursos y servicios externos tienen un menor coste que los producidos en nuestro sistema.

Los cambios en los requisitos y las condiciones, tanto internas como externas al sistema, producirán la necesidad de cambiar la funcionalidad del sistema, con lo que un proceso de adaptación será necesario para conseguir que el sistema desarrolle la funcionalidad deseada [APDD08].

## B. Proceso de adaptación

El proceso de adaptación debe producirse haciendo cambios directamente en la organización. Este proceso puede darse de varias formas, dependiendo siempre del tipo de cambios que podría sufrir el sistema. Estos cambios pueden ser predefinidos, controlados y emergentes, además de existir un proceso genérico para realizar estos cambios [PHBG09].

El proceso genérico de adaptación de la organización se puede dividir en dos partes [PHBG09]: monitorización y reparación. La segunda fase puede, a su vez, ser descompuesta en las fases de diseño, selección y ejecución, o únicamente en selección y ejecución. La fase de monitorización se basa en encontrar problemas en el sistema, la organización y su entorno. Es recomendable definir un conjunto de situaciones de no-adaptación, para que cuando sean detectadas el sistema pueda responder a esas situaciones. Una vez se detecta la necesidad de adaptación es necesario diseñar un conjunto de alternativas que llevar a cabo en la organización, seleccionar la alternativa más adecuada para la organización, con lo que es necesario definir un criterio de evaluación para decidir la alternativa más adecuada, y finalmente poner en práctica esa solución. Este es el proceso general para realizar la adaptación de un SMA, pero existen otros procesos para realizar la adaptación más específicos que modifican el proceso genérico y son definidos a continuación:

- **Cambio predefinido en la organización** [PHBG09]: este tipo de cambios se encuentra previamente planificado para que ocurra en un momento determinado. La monitorización la llevan a cabo los agentes o una entidad externa, la fase de diseño se realiza previamente en tiempo de diseño del SMA y las fases de selección y ejecución se ejecutan de inmediato al darse el momento del cambio.

- **Cambio controlado en la organización** [PHBG09]: el sistema no sabe cuándo cambiará la organización, pero conoce las condiciones que propiciarán el cambio. Es necesario definir estrategias para monitorizar la organización, ya sea de forma endógena (por los propios agentes) o exógena (el usuario o un sistema externo controlan estos cambios). Se debe conocer cómo identificar que el objetivo global no se cumple porque no está permitido por la actual organización y qué parte de la organización obliga a hacer el cambio. Una vez identificado el problema, es necesario diseñar un conjunto de alternativas, que pueden venir dadas en un conjunto predefinido o por el contrario pueden ser diseñadas a petición del usuario, con el correspondiente problema de la complejidad surgida al tener un espacio tan grande de soluciones en el que buscar.
- **Cambio dirigido por el *system adapter***: en este caso, ni el diseñador del sistema ni los agentes conocerán de antemano las condiciones y el momento para hacer el cambio en una organización virtual. Por tanto, no es necesario que éstos realicen una monitorización del sistema. Será el *system adapter* el que dé la orden para iniciar el proceso de cambio dentro del sistema, siendo el responsable de conocer las condiciones y el momento de cambiar, además de monitorizar el sistema. Una vez dada la orden de reorganización, el *system adapter* dará las claves del cambio, que podrá ser de uno de los siguientes tipos:
  - El *system adapter* decide de manera unívoca el cambio que debe producirse en la organización. Los agentes de la organización no tienen opción de rebatir este cambio y la modificación se produce.
  - El *system adapter* da varias opciones de cambio, con lo que es necesaria la fase de selección para elegir la opción más beneficiosa para la organización.
  - El *system adapter* proporciona únicamente una serie de guías, consejos, etc. que debe seguir la organización y serán los agentes los que diseñen el conjunto de alternativas, teniendo después el *system adapter* que seleccionar la más adecuada y aplicarla.
- **Cambio emergente de la organización** [PHBG09]: Los cambios surgen a nivel de agente, no de organización. La nueva organización surge de las interacciones que realizan los agentes. Los agentes conocen cuándo la organización no es conveniente y producen cambios de forma local, hecho que propicia el cambio emergente a nivel global.

Uno de los factores a tener en cuenta cuando se desarrollan sistemas adaptativos es el momento en el que se deben contemplar los cambios que el

sistema podría tener que llevar a cabo. Existirán dos momentos en los que los cambios deberán ser tenidos en cuenta: en fase de diseño o de ejecución.

Cabe la posibilidad de que en fase de diseño seamos capaces de conocer los cambios que el sistema vaya a tener que afrontar durante su ejecución. Se hace necesaria la inclusión de guías en una metodología de desarrollo de sistemas multiagente que nos indiquen los pasos a seguir para realizar procesos de adaptación. Estas guías (que podrían estar basadas en los patrones de diseño [GVO07] definidos en el framework Agents & Artifacts [RVO07]) ayudarán en la identificación de las situaciones que deben ser tenidas en cuenta de antemano para adaptar el sistema. Estas situaciones serán las anteriormente comentadas, los cambios predefinidos y controlados del sistema. En el primer caso, las guías tendrán en cuenta cuándo y cómo reorganizar y en el segundo caso únicamente se especifican las condiciones para la reorganización, sin especificar qué hacer una vez detectadas esas condiciones.

En fase de ejecución el sistema debe ser capaz de conseguir gestionar de forma correcta tanto los cambios debidos al *system adapter* como los cambios emergentes. Es decir, debemos contar con agentes que hayan sido desarrollados pensando en tener que soportar cambios en la estructura de la organización, ya sea para monitorizar la organización, diseñar las posibles estrategias o para ponerlas en funcionamiento. Para ello, deben estar provistos de comportamientos que soporten adaptación, con lo que podría ser necesario modificar la arquitectura de los agentes que pueblan un SMA no adaptativo.

#### 2.6.4. Discusión

En este capítulo se ha presentado una visión general del estado del arte en torno a la adaptación en los Sistemas Multiagente. Concretamente, al estar este trabajo apoyado en los OCMAS, nos hemos centrado en estudiar el cambio de la estructura organizativa de un SMA. Estos cambios pueden venir dados de dos formas. La primera es la auto-organización, un enfoque *bottom-up*, en el que las interacciones entre agentes de manera local llevarán a conseguir el comportamiento deseado del sistema a nivel global. El segundo enfoque es la reorganización, una visión *top-down*, donde los cambios en la estructura organizativa (normas, roles, topología, etc.) provocarán modificaciones en las interacciones entre agentes.

En estos dos enfoques se han realizado diferentes trabajos, siendo más numerosos los trabajos de auto-organización, muchas veces basados en procesos de la naturaleza como los caminos seguidos por las hormigas. Por otra parte, la reorganización es un enfoque que todavía no ha sido estudiado con detalle, con lo que existen menos aportaciones en este tema, y sigue siendo

un importante campo de estudio a la hora de conseguir que un SMA sea capaz de soportar la adaptación.





## Capítulo 3

# Modelado de mecanismos organizativos empleando artefactos

Este capítulo describe cómo los dos tipos de mecanismos organizativos, informativos y regulativos, presentados en el capítulo 2, sección 2.5, pueden ser modelados como artefactos. Los artefactos permiten una sencilla fusión de los mecanismos organizativos dentro del entorno de un SMA.

Debido al cambio en los SMA desde una concepción de sistema cerrado, en donde el comportamiento de los agentes es predecible, a una percepción de sistema abierto, donde existen agentes externos, posiblemente con comportamientos interesados, aumenta la necesidad de contar con mecanismos que sean capaces de mejorar la coordinación y el comportamiento de los agentes para alcanzar de forma más efectiva los objetivos de la organización.

Como ya se explicó anteriormente, un método para mejorar el comportamiento de la organización son los mecanismos organizativos, que presentan una forma efectiva de influir en el comportamiento de los agentes del sistema. Estos mecanismos podrán estar ubicados en el entorno del sistema multiagente, al igual que los artefactos, presentados en el capítulo 2, sección 2.4. Además, una de las funciones para las que podemos usar artefactos es para mejorar la coordinación entre los miembros del sistema, al igual que los mecanismos organizativos. Por tanto, el objetivo de este capítulo es presentar un modelado de los mecanismos organizativos utilizando artefactos, dando lugar a un nuevo tipo de artefactos, los *artefactos para mecanismos organizativos*.

El resultado de este modelado es un conjunto de tres tipos de artefactos [ECHA10]. Los *artefactos informativos* están basados en los mecanismos informativos; los *artefactos incentivos* están basados en los mecanismos incen-

tivos, mientras que los *artefectos coactivos* están basados en los mecanismos coactivos.

### 3.1. Conceptos básicos

Antes de presentar el modelado de los mecanismos organizativos por medio de artefactos, será necesario presentar una serie de definiciones de distintos conceptos.

En primer lugar, debemos definir un sistema multiagente de forma que sea capaz de albergar artefactos y mecanismos organizativos. Los autores de los mecanismos organizativos [CBHO09] definen un SMA de la siguiente manera:

**Definición 1.** *Un Sistema Multiagente (SMA) es una tupla  $\langle \mathcal{Ag}, \mathcal{A}, \mathcal{X}, \Phi, x_0, \varphi \rangle$  donde:*

- $\mathcal{Ag}$  es un conjunto de agentes;  $|\mathcal{Ag}|$  denota el número de agentes del sistema;
- $\mathcal{A}$  es un espacio de acciones finito que incluye todas las posibles acciones que pueden llevarse a cabo en el sistema.  $\mathcal{A}$  incluye la acción  $a_{skip}$ , la acción de no hacer nada;
- $\mathcal{X}$  es el espacio de estados del entorno;
- $\Phi : \mathcal{X} \times \mathcal{A}^{|\mathcal{Ag}|} \times \mathcal{X} \rightarrow [0..1]$  es la distribución de probabilidad de transición del SMA, describiendo cómo el entorno evoluciona como resultado de las acciones de los agentes;
- $x_0 \in \mathcal{X}$  es el estado inicial del SMA;
- $\varphi : \mathcal{Ag} \times \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}$  es la función de capacidad de los agentes que describe las acciones que los agentes son capaces de realizar en un estado dado del entorno.  $\varphi(a, x, ac) = 1$  ( $\varphi(a, x, ac) = 0$ ) significa que el agente  $a$  es capaz (no es capaz) de realizar la acción  $ac$  en el estado  $x$ .

Un artefacto es un elemento de un SMA que los agentes utilizan para conseguir sus objetivos. De manera formal, un artefacto se define como:

**Definición 2.** *Un artefacto es una tupla  $\langle PR, OP, LO, St \rangle$  donde:*

- $PR$  son las propiedades observables del artefacto que los agentes pueden comprobar directamente sin invocar una operación;

- *OP* es el conjunto de operaciones que los agentes pueden ejecutar cuando interactúan con ellos;
- *LO* se refiere a las operaciones de enlace, que pueden ser llamadas por otros artefactos. Este tipo de operaciones permite la composición de artefactos y la distribución de funcionalidad uniendo artefactos. En algunos casos, estas operaciones pueden ser usadas para ayudar en la inicialización de otro artefacto;
- *St* es el estado interno de un artefacto, que no es accesible por los agentes que habitan el sistema.

Respecto a los mecanismos organizativos, disponemos de las siguientes definiciones [CBHO09]:

**Definición 3.** *Un mecanismo organizativo informativo es una función que proporciona información, dadas las descripciones, posiblemente parciales, del entorno del sistema y del agente que solicita información. Formalmente se define como:*

$$\Gamma : \mathcal{S}' \times \mathcal{X}' \rightarrow \mathcal{I}$$

donde:

- $\mathcal{S}'$  representa el conjunto de descripciones parcialmente posibles del estado interno de los agentes;
- $\mathcal{X}'$  es el conjunto de vistas parciales de los estados del entorno;
- $\mathcal{I}$  representa un espacio de información.

**Definición 4.** *Un mecanismo organizativo incentivo es una función que produce cambios en la distribución de probabilidad de transición del SMA, dada una descripción de su entorno. Formalmente:*

$$\Upsilon_{inc} : \mathcal{X}' \rightarrow [\mathcal{X} \times \mathcal{A}^{|\mathcal{A}g|} \times \mathcal{X} \rightarrow [0..1]]$$

**Definición 5.** *Un mecanismo organizativo coactivo es una función que produce cambios en la función de capacidad de los agentes, dada una descripción parcial del entorno del SMA. Formalmente:*

$$\Upsilon_{coe} : \mathcal{X}' \rightarrow [\mathcal{A}g \times \mathcal{X} \times \mathcal{A} \rightarrow \{0,1\}]$$

donde  $\mathcal{X}'$  representa el conjunto de descripciones posiblemente parciales de los estados del entorno de un SMA.

## 3.2. Artefactos de mecanismos organizativos

A fin de facilitar el diseño e implementación de los mecanismos organizativos, los hemos modelado empleando el concepto de artefacto. A continuación se explicarán los tres tipos de artefactos definidos: *informativos*, *incentivos* y *coactivos*.

### 3.2.1. Artefactos informativos

Los mecanismos informativos devuelven información sobre acciones a un agente, dada una descripción parcial de su estado interno y teniendo en cuenta la vista parcial del entorno que tiene el mecanismo. Hemos modelado el mecanismo informativo como un artefacto, llamado *artefacto informativo*, siendo una entidad pasiva usada por los agentes para ayudarles en su proceso deliberativo.

**Definición 6.** *Un artefacto informativo se define como un artefacto  $Ar_{inf} = \langle PR, OP, LO, St \rangle$  donde:*

- $PR \subseteq \{St \cup \emptyset\}$  son las propiedades observables del artefacto informativo, que son un subconjunto de la información contenida dentro del artefacto o un conjunto vacío.
- $OP : S' \rightarrow \mathcal{I}$  son las operaciones del artefacto, donde:
  - $S'$  representa una descripción parcial del estado interno de un agente.
  - $\mathcal{I}$  representa la información devuelta por el artefacto, basada en el estado interno del artefacto y la descripción parcial del estado interno del agente (semánticamente,  $S' \times St \rightarrow \mathcal{I}$ ).
- $LO : \Theta \rightarrow \mathcal{I}$  es una operación de enlace usada por un artefacto  $Ar_1$  para obtener información del artefacto  $Ar_{inf}$ , donde:
  - $\Theta \subseteq (\Sigma \cup S')$  es la información enviada por  $Ar_1$  a  $Ar_{inf}$ ;
  - $\Sigma \subseteq \{St_1 \cup \emptyset\}$  es el estado parcial de  $Ar_1$ , siendo  $St_1$  la información interna de  $Ar_1$ ;
  - $S'$  representa una descripción parcial del estado interno del agente que solicita la información al artefacto  $Ar_1$ ;
  - $\mathcal{I}$  representa la información devuelta por el artefacto  $Ar_{inf}$  al artefacto  $Ar_1$ , basada en la descripción parcial de  $Ar_1$  ( $\Sigma$ ), la descripción parcial del estado interno del agente que está solicitando

a  $Ar_1(S')$  y el estado interno del artefacto  $Ar_{inf}(St)$ . Semánticamente:  $(\Sigma \cup S') \times St \rightarrow \mathcal{I}$ .

- $St$  representa el estado interno del artefacto, por ejemplo la información contenida en el artefacto, que no es directamente accesible por los agentes u otros artefactos.

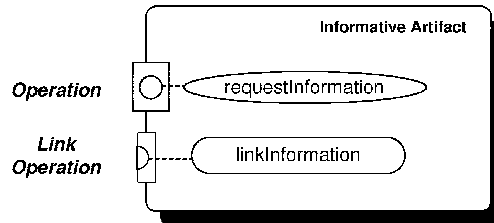


Figura 3.1: Artefacto informativo para un mecanismo informativo

La figura 3.1 muestra una representación de un artefacto informativo. Como se ha explicado, este tipo de artefacto necesita al menos una operación: `requestInformation(St)`. Dando una descripción parcial del estado interno del agente, que puede contener los roles, las creencias, los hechos y otras características asociadas con el agente y su entorno, esta operación devuelve un paquete de información que contiene: (i) el tipo de información, que puede ser una recomendación o un aviso sobre acciones, o información sobre las consecuencias de ejecutar una acción; (ii) la descripción de esta información, y (iii) un conjunto de acciones que están relacionadas con esta información (podrían ser servicios que un agente puede ofrecer, servicios recomendados, servicios que tienen consecuencias en el agente, etc.). Además, como se ha explicado, un artefacto informativo podría necesitar realizar peticiones a otros artefactos para obtener información o actualizar su estado interno. Por este motivo, los artefactos informativos se mejoran con una operación de enlace, `linkInformation(S', St)`, que se ejecuta como consecuencia de una operación ( $OP$ ) invocada en otro artefacto. El artefacto que hace la petición,  $Ar_1$ , envía el estado parcial del agente (estado interno del agente) que solicitó información, además de un subconjunto de su información interna. Como resultado, el artefacto al que se le ha solicitado información,  $Ar_{inf}$ , devolverá extractos de información basados también en su propio estado interno.

No resulta indispensable que los artefactos informativos implementen operaciones de enlace, con lo que sólo podrían ser accesibles en su mismo workspace. Cuando ofrecen una operación de enlace, los artefactos localizados en su mismo workspace o en otros workspaces conectados pueden obtener información relevante de este artefacto informativo por medio de sus operaciones de enlace.

Para ejemplificar el funcionamiento de este tipo de artefactos, hemos diseñado un artefacto que proporciona las normas activas actualmente en el sistema. Este artefacto contendría las operaciones mostradas en la figura 3.2.

Sea  $Ar_{inf}^{norms} = \langle \emptyset, \{requestNorms\}, \{linkInformation\} \rangle$  un artefacto dirigido a proporcionar información (bajo demanda) a los agentes sobre normas, como la especificación de normas que manejan un rol, normas activas o desactivadas, etc. La operación *linkInformation* puede ser usada por otros artefactos para recolectar información relacionada con las normas que pueden mejorar su uso. En consecuencia, el artefacto proporciona información sobre las normas que controla tanto a agentes, que harán un uso personal de ella, como a otros artefactos.

De este modo, el artefacto encapsula la funcionalidad para los agentes que solicitan información para su uso personal, y otros artefactos en los que podrían estar interesados en alguna información sobre normas que el artefacto controla.

Se puede comprobar que no existe ninguna propiedad observable, ya que las normas no pueden ser directamente accesibles por los agentes, pero pueden ser solicitadas usando la operación *requestNorms*. Considerando esto, debido a que es un artefacto informativo, el agente solicitante debe enviar una parte de su estado mental para permitir al artefacto devolverle información útil.

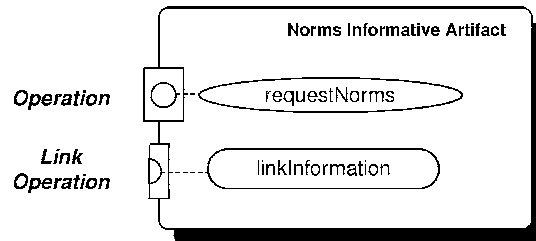


Figura 3.2: Artefacto informativo para informar sobre normas

Cabe destacar que este artefacto no es un mero repositorio de normas, ya que permite ser modificado para distinguir entre diferentes tipos de información que debe ser proporcionada a los agentes. En consecuencia, el diseñador del mecanismo probablemente no quiera que cualquier agente pudiese conocer todas las normas en cualquier momento, pero posiblemente prefiera dar información precisa al agente que esté interesado en ella, de tal manera que no revele información sensible.

Un escenario típico consistiría en un agente solicitando el conjunto de normas que controlan un rol específico que el agente quiere jugar. Las responsabilidades, deberes y derechos de un rol es un tipo de información que el artefacto debe proporcionar a petición.

### 3.2.2. Artefactos incentivos

Como se ha explicado anteriormente, los artefactos incentivos son mecanismos capaces de producir cambios en el entorno del sistema desde una perspectiva global, modificando las recompensas y sanciones que están activas en el sistema. Estos mecanismos se basan en la creencia de que un cambio posiblemente pequeño en el sistema de incentivos (que puede afectar sólo a un pequeño número de agentes) afecta al sistema completo. En esta subsección, los mecanismos de incentivos se modelan como artefactos, llamados *artefactos incentivos*. Estos artefactos ejecutarán cambios organizativos, que brindan la posibilidad de implementar un sistema adaptativo, variando los elementos del sistema (por ejemplo, añadiendo o eliminando normas). Después de que se produzca un cambio en el sistema de incentivos del SMA se ven afectadas las probabilidades de transición entre diferentes estados del sistema. Para llevar a cabo estos cambios, es necesario tener un agente o un humano jugando un rol especial que llamamos '*system adapter*', que es capaz de controlar los cambios organizativos cuando sea necesario para promover la capacidad de adaptación del SMA. El '*system adapter*' es el único agente con privilegios para ejecutar las operaciones de un artefacto incentivo.

**Definición 7.** *Un artefacto incentivo se define como un artefacto  $Ar_{inc} = \langle PR, OP, LO, St \rangle$  donde:*

- $PR \subseteq \{St \cup \emptyset\}$  son sus propiedades observables;
- $OP : \Delta$  es la operación que permite al system adapter introducir o eliminar incentivos en el sistema;
- $LO = \emptyset$ , debido a que este tipo de artefactos no tiene operaciones de enlace predefinidas;
- $St$  representa el estado interno del artefacto.

La operación del artefacto ( $OP$ ) modifica la probabilidad de transición entre diferentes estados del sistema. Esta operación se define como:

$$\Phi = St \rightarrow [\mathcal{X} \times \mathcal{A}^{|\mathcal{A}g|} \times \mathcal{X} \rightarrow [0..1]], \text{ donde:}$$

- $\Phi$  es la distribución de probabilidad de transición de un SMA, describiendo cómo el entorno evoluciona como resultado de las acciones de los agentes.
- $\mathcal{X}$  es el espacio de estados del entorno.

- $\mathcal{A}^{[Ag]}$  es el conjunto de acciones ejecutadas por los agentes entre dos estados del SMA.

Esta operación funciona así: el agente proporciona información al artefacto, que podría cambiar su estado interno ( $St$ ). Dado este nuevo estado interno, se modifica la probabilidad de transición entre dos estados del sistema, por lo que el comportamiento del SMA cambia desde una perspectiva global.

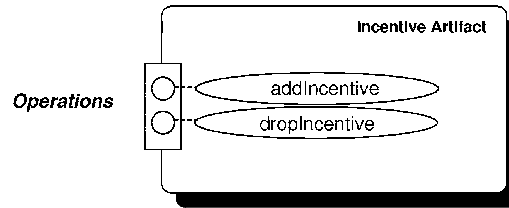


Figura 3.3: *Artefacto incentivo*

La figura 3.3 muestra cómo se modela un artefacto incentivo, incluyendo sus características mínimas requeridas. El conjunto  $OP$  contiene dos operaciones distintas: **addIncentive**( $\{ty, ro, ac\}$ ), (con  $\{ty, ro, ac\} \in St$ ), que añaden un incentivo al sistema de incentivos del SMA; y **dropIncentive**( $\{ty, ro, ac\}$ ) (con  $\{ty, ro, ac\} \in St$ ), que elimina un objetivo del SMA. En estas funciones  $ty$  significa el tipo de incentivo (recompensa o sanción),  $ro$  se refiere al rol o al conjunto de roles afectados por esta iniciativa y  $ac$  representa el conjunto de acciones relacionadas con este incentivo.

A veces, podría ser útil enviar información sobre cambios en el sistema de incentivos de los agentes que habitan el SMA. Para ejecutar esta tarea es necesario proporcionar al entorno un artefacto informativo, modelado como se ha explicado en la subsección anterior.

Como ejemplo de artefactos incentivos, se ha elegido de nuevo un entorno organizativo relacionado con las normas.

Sea  $Ar_{inc}^{norms} = \langle \emptyset, \{addNormIncentive, dropNormIncentive\}, \emptyset \rangle$  un artefacto incentivo que permite introducir incentivos positivos (*recompensas*) e incentivos negativos (*sanciones*) dentro de una organización. Estos incentivos consisten en un conjunto de posibles consecuencias que pueden implicar el cumplimiento o violación de normas. Como se ha citado, los mecanismos de incentivos pretenden mejorar el rendimiento del sistema introduciendo cambios en el entorno que de alguna manera influyan en el razonamiento de los agentes. Para este ejemplo consideramos que el artefacto no contiene ninguna propiedad observable y que no ofrece ninguna operación de enlace que pueda ser solicitada por otros artefactos. La interfaz de uso ( $OP$ ) podría no estar disponible para cada agente participante del sistema. Por tanto, este



tipo de artefactos no proporciona información, pero cambia el entorno, por lo que sólo los agentes con los permisos requeridos para hacerlo pueden usar las operaciones contenidas en  $OP$ , dependiendo del dominio. En nuestro caso los agentes capaces de jugar el rol 'system adapter' pueden emplear la operación  $addNormIncentive$ , por lo que añaden una sanción a una norma en caso de violación; o introduciendo recompensas en el caso del cumplimiento de normas. Los incentivos pueden ser actualizados en el tiempo, usando la operación  $dropIncentive$  para eliminar los antiguos y la operación  $addNormIncentive$  para añadir nuevos incentivos.

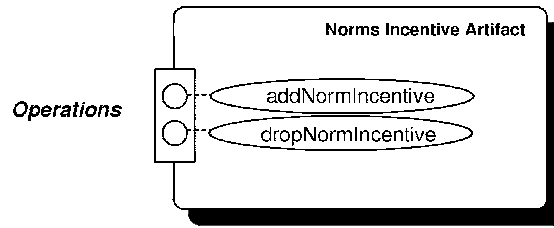


Figura 3.4: Artefacto informativo para normas

### 3.2.3. Artefactos coactivos

Como se ha explicado anteriormente, los mecanismos coactivos están centrados en producir cambios en el entorno del sistema produciendo cambios en las funciones de capacidad de los agentes, dada una descripción del SMA posiblemente parcial. Como ocurre con los mecanismos incentivos, los mecanismos coactivos también confían en la existencia de un rol 'system adapter', que es capaz de promover cambios organizativos.

Formalmente, un artefacto coactivo se define como:

**Definición 8.** Un *artefacto coactivo* es un artefacto  $Ar_{coe} = \langle PR, OP, LO, St \rangle$  donde:

- $PR \subseteq \{St \cup \emptyset\}$  son sus propiedades observables;
- $OP : St \rightarrow [Ag \times \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}]$  es la operación llevada a cabo por el artefacto coactivo, donde:
  - $Ag$  es un agente del SMA;
  - $\mathcal{A}$  es el espacio de acciones que incluye todas las posibles acciones que pueden ser llevadas a cabo en el sistema.

- $LO = \emptyset$ , ya que este tipo de artefactos no tiene operaciones de enlace predefinidas;
- $St$  representa el estado interno del artefacto.

La operación  $St \rightarrow [\mathcal{A}g \times \mathcal{X} \times \mathcal{A} \rightarrow \{0, 1\}]$ , dado el estado interno del artefacto, devuelve la capacidad para ejecutar o no una acción, 1 y 0 respectivamente. Internamente, esta operación trabaja así: el artefacto necesita su estado interno ( $St$ ) así como la información proporcionada por el 'system adapter' ( $\mathcal{A}g$  y  $\mathcal{A}$ ) para ejecutar esta operación. Después de recopilar toda esta información, el artefacto calcula el nuevo espacio de acciones del agente. Este cambio puede ser visto como un cambio local pero dado que los agentes están interrelacionados, los cambios en un único agente podrían producir cambios en un conjunto de agentes, es decir, en el estado global del SMA.

De manera similar a los artefactos informativos e incentivos, los artefactos coactivos no tienen obligatoriamente propiedades observables. En caso de tenerlas, son un subconjunto de conocimiento interno del artefacto. El número de las propiedades observables disponibles dependerá del propósito de cada artefacto. Además, no se requiere que tengan operaciones de enlace.

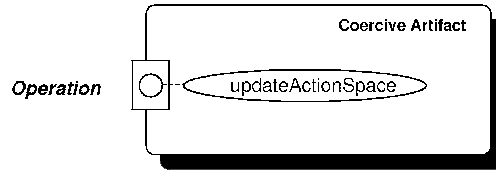


Figura 3.5: Artefacto coactivo.

La Figura 3.5 muestra cómo se modela un artefacto coactivo, incluyendo sus requisitos mínimos. La única operación definida en este artefacto es  $updateActionSpace(\{ag, a\})$  (donde  $\{ag, a\} \in St$  y  $ag \in \mathcal{A}g, a \in \mathcal{A}$ ), que recibe un agente y una acción del 'system adapter' y devuelve la capacidad del agente dado para realizar esta acción.

Como se hizo con los artefactos incentivos, para mostrar un ejemplo de artefactos coactivos, se toma en cuenta un entorno organizativo relacionado con las normas:

Sea  $Ar_{coe}^{norms} = \langle \emptyset, \{updateActionSpace\}, \emptyset \rangle$  un artefacto coactivo centrado en actualizar el espacio de acciones de los agentes durante el tiempo. Como se ha dicho en la sección 2.5, los mecanismos coactivos modifican directamente los espacios de acciones de los agentes para mantenerlos alejados de comportamientos indeseables. En consecuencia, este artefacto estará a cargo de modificar estos espacios de acciones bajo demanda de algunos agentes especiales que tienen permiso para introducir estos cambios en el entorno. Por

lo tanto, si uno de estos agentes con permisos suficientes (es decir, el *'system adapter'*) observa que, por ejemplo, una norma ha sido violada, puede tomar la decisión de prohibir ciertas acciones al agente que no cumplía esa norma, intentando evitar ese comportamiento en el futuro. De la misma manera que el artefacto puede eliminar acciones del espacio de acciones de los agentes, también puede añadir acciones si el comportamiento del agente está siendo aceptable. Por ejemplo, el sistema puede examinar a los participantes con un periodo de prueba para asegurarse que se comparta de acuerdo a los objetivos del sistema, permitiéndoles realizar más y más acciones progresivamente.

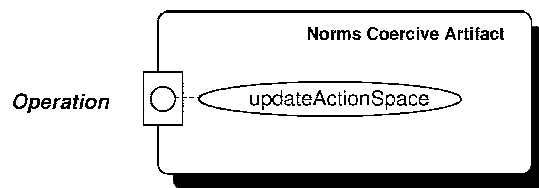


Figura 3.6: Artefacto coactivo para normas

Algunos ejemplos de mecanismos que podrían ser diseñados como artefactos regulativos o incentivos son: un controlador normativo, que encapsula las consecuencias que puede implicar el cumplimiento o la violación de normas; o el controlador de multas de tráfico, donde diferentes sanciones sobre normas de circulación pueden aplicarse, incluso introduciendo restricciones en el entorno (se pueden cortar carreteras, retirar permisos de conducir, etc.).

### 3.3. Trabajo relacionado

Esta sección se centra en comparar nuestra propuesta con otros artefactos disponibles presentados por diferentes autores, para determinar si los artefactos existentes tienen las características de artefactos informativos, incentivos o coactivos.

Los artefactos informativos proporcionan una funcionalidad común para un SMA. Algunos de los artefactos presentados por la comunidad de investigadores proporcionan información a otros agentes después de recibir información sobre una vista parcial de su estado interno, por lo que pueden ser vistos como artefactos informativos. Por ejemplo, el *Role Evolution Coordination Artifact* [HBO10], es un artefacto destinado a construir y evolucionar una taxonomía de especialización de roles, la cual consiste en un conjunto de roles con un orden determinado sobre el tiempo; y poner esta información a disposición de los agentes. Este artefacto contiene tres operaciones: (i) *getBestRolesForInteraction*, que proporciona los roles más especializados para

una interacción de tipo de servicio dada; (ii) *getAgentsForRoles*, que proporciona el conjunto de agentes que juegan al menos uno de los roles de un conjunto dado de roles; y (iii) *getRolesForAgent*, que proporciona el conjunto de roles que un agente dado juega en el sistema.

Anteriormente, en la sección 3.2.1, hemos definido formalmente la operación de un artefacto informativo como  $\mathcal{S}' \times \mathcal{St} \rightarrow \mathcal{I}$ . Es posible establecer una correspondencia entre las operaciones del *Role Evolution Coordination Artifact* y la operación de un artefacto informativo. De esta forma, la función de *getBestRolesForInteraction* puede describirse como:

- $\mathcal{S}' = \text{Serv}$ , donde *Serv* es una interacción de tipo de servicio.
- $\mathcal{St} = R$ , donde  $R$  es el conjunto completo de roles del SMA.
- $\mathcal{I} = \mathcal{P}(R)$ , donde  $\mathcal{P}(R)$  son los roles más especializados para  $S$ .

De forma similar, la función *getAgentsForRoles* tiene la siguiente correspondencia:

- $\mathcal{S}' = \mathcal{P}(R)$ , donde  $\mathcal{P}(R)$  es un conjunto de roles.
- $\mathcal{St} = \mathcal{Ag}$ , donde  $\mathcal{Ag}$  es el conjunto completo de agentes del SMA.
- $\mathcal{I} = \mathcal{P}(\mathcal{Ag})$ , donde  $\mathcal{P}(\mathcal{Ag})$  es el conjunto de agentes que juegan al menos uno de los roles de  $\mathcal{P}(R)$ .

Finalmente, la función *getRolesForAgent* presenta la siguiente correspondencia con una operación de un artefacto informativo:

- $\mathcal{S}' = \mathcal{Ag}$ , donde  $\mathcal{Ag}$  es un agente del sistema.
- $\mathcal{St} = R$ , donde  $R$  es el conjunto completo de roles del sistema.
- $\mathcal{I} = \mathcal{P}(R)$ , donde  $\mathcal{P}(R)$  es el conjunto de roles que  $\mathcal{Ag}$  juega en el sistema.

Otro ejemplo de artefacto que puede ser considerado como artefacto informativo es el *Co-Argumentation Artifact* (CAA) [OMO08], que ayuda en los procesos de argumentación. Los agentes comparten sus argumentos (es decir, una vista parcial de su estado interno) con el artefacto. Después, el artefacto evalúa los argumentos proporcionados por todos los agentes y calcula la *aceptación social* (la aceptación de los argumentos de un agente concreto) y el *comportamiento social* (la aceptación de los argumentos desde una perspectiva global).

La implementación de un CAA propuesta en [OMO08] proporciona dos propiedades observables (*Social Behavior*, *Social Acceptability*) y una operación (*writeArguments*), que permite a los agentes almacenar sus argumentos en el artefacto.

Siguiendo la formalización de artefactos para mecanismos organizativos, este CAA puede ser modelado como un artefacto informativo o bien como un artefacto incentivo. En este caso, este artefacto puede ser implementado con dos operaciones distintas: *getSocialValues* y *writeArguments*.

Es posible establecer una correspondencia entre la operación *getSocialValues* de un Co-Argumentation Artifact y la operación requerida de un artefacto informativo. La descripción parcial del estado interno de un agente ( $\mathcal{S}'$ ) se representa en un CAA como el argumento que un agente usará durante el proceso de argumentación.

$$\mathcal{S}' = Arg_t$$

donde  $Arg_t$  es un argumento proporcionado por el agente  $t$ .

En este ejemplo, el estado interno del artefacto ( $St$ ) es el conjunto de argumentos que han sido almacenados hasta el momento.

$$St = \bigcup_{i=1}^n Arg_i$$

donde  $St$  es la compilación de  $n$  argumentos.

Finalmente, la información ( $\mathcal{I}$ ) devuelta por el artefacto son los valores de Aceptación Social y de Comportamiento Social.

$$\mathcal{I} = \{SocAcc, SocBeh\}$$

donde *SocAcc* es la Aceptación Social y *SocBeh* es el Comportamiento Social.

La operación *writeArguments* puede ser empleada para establecer un mecanismo incentivo. En este caso, esta función sólo puede ser usada por un agente jugando el rol de *system adapter*, que puede tomar ventaja de este artefacto controlando qué argumentos propuestos por los agentes tienen que ser almacenados dentro del artefacto para promover un comportamiento concreto hacia el objetivo global del sistema. De esta forma, sólo aquellos argumentos que pueden ayudar a promover este comportamiento esperado se almacenarán usando la operación *writeArguments*.

Como se ha explicado en la sección 3.2.2, la operación de un artefacto incentivo implica:

$$\Phi = St \rightarrow [\mathcal{X} \times \mathcal{A}^{|Ag|} \times \mathcal{X} \rightarrow [0..1]]$$

Además, esta operación modifica el estado interno del artefacto ( $St$ ), que produce un cambio en las transiciones de la función de utilidad global del

sistema. La figura 3.7 muestra cómo funciona este proceso. Por ejemplo, el Agente A y el Agente B proponen sus argumentos,  $Arg_A$  y  $Arg_B$ , respectivamente. El system adapter recibe estos argumentos y decide que  $Arg_A$  ayuda de mejor manera a conseguir el comportamiento social esperado, por lo que emplea la operación *writeArguments* para guardar este argumento en el artefacto, aplicando así un incentivo.

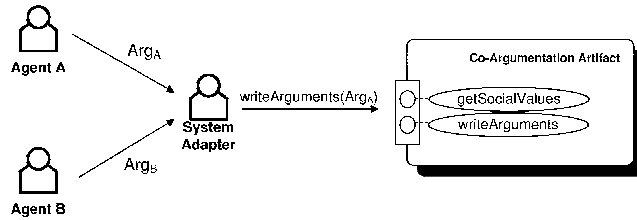


Figura 3.7: Un CAA trabajando como artefacto incentivo

De manera similar, otros tipos de artefactos, como los *Coordination artifacts* [ORV<sup>+</sup>04], los *Organizational artifacts* [HBKR09] o los *Reputation Artifacts* [HBV08], también pueden ser descritos con características de mecanismos organizativos. Por tanto, ya que los artefactos de coordinación encapsulan un servicio de coordinación, este servicio de coordinación puede ser implementado por medio de un artefacto informativo (proporcionando información útil a los agentes), un artefacto incentivo (modificando la probabilidad de transición entre diferentes estados del sistema) o un artefacto coactivo (permitiendo o prohibiendo a los agentes desarrollar diferentes acciones). Con respecto a los artefactos organizativos, son usados para controlar una organización de agentes ayudando a la organización a alcanzar sus metas desde un nivel global y social. Un ejemplo claro de este tipo de artefactos es un artefacto que ayude a informar o a controlar normas, el cual, como se ha explicado en la sección 3.1, puede ser modelado como un artefacto informativo (proporcionando las normas actualmente activas en el sistema), un artefacto incentivo (introduciendo incentivos positivos o negativos dentro de la organización) o un artefacto coactivo (eliminando acciones del espacio de acciones del agente o incluyendo nuevas acciones posibles). Finalmente, los artefactos de reputación encapsulan la colección de violaciones de normas de los participantes en un sistema y después las unen para permitir a los agentes consultar la reputación usando las propiedades observables de los artefactos.

Como se ha mostrado en esta sección, las características importantes de muchos artefactos existentes pueden ser modeladas siguiendo la formalización propuesta de artefactos para mecanismos organizativos. En la mayoría de casos, las operaciones ofrecidas actualmente por los artefactos hacen que sean fácilmente modelados como artefactos informativos. Por otra parte, la

formalización propuesta puede ser también útil para extender las operaciones de los artefactos actuales para aplicar mecanismos incentivos y/o coactivos dentro del entorno.

### 3.4. Discusión

Los Mecanismos Organizativos se centran en mejorar la coordinación entre los agentes de un SMA, intentando cambiar esta coordinación desde una micro-perspectiva (es decir, la perspectiva de agentes individuales), proporcionando información útil a los agentes (mecanismos informativos); y una macro-perspectiva (es decir, la perspectiva del SMA completo), modificando las consecuencias de las acciones (mecanismos incentivos) o las funciones de capacidad de los agentes (mecanismos coactivos).

En este trabajo, estos mecanismos se han modelado como artefactos para facilitar a los desarrolladores un mejor despliegue e implementación de los mismos, así como para añadir funcionalidad a los entornos de los SMA. Se han definido tres tipos de Artefactos para Mecanismos Organizativos: (i) Artefactos Informativos, que proporcionan información a un agente basada en el estado interno de este agente y la vista parcial del entorno que tiene el artefacto; (ii) Artefactos Incentivos, que modifican el comportamiento global del sistema cambiando el sistema de incentivos del SMA; y (iii) Artefactos Coactivos, que actualizan el espacio de acciones de un agente. Todos estos artefactos hacen uso del entorno de un SMA, por lo que pueden explotar todo el conocimiento que tienen sobre las entidades que pueblan el sistema.





# Capítulo 4

## Extensión del Virtual Organization Model

La sección 2.2.1 de este trabajo dió una visión global de un lenguaje de modelado de organizaciones (OML) determinado, el denominado Virtual Organization Model (VOM). Este capítulo se centrará en presentar de forma más detallada este modelo, además de citar las modificaciones que se han realizado para la presente tesis de máster, con el objetivo de mejorar la expresividad y funcionalidad del metamodelo, presentado originalmente por Estefanía Argente en su tesis doctoral [Arg08].

En la figura 4.1, presentada en la sección 2.2.1 se muestra una visión general del VOM, mostrando con colores los principales elementos que componen el metamodelo. En ella se puede comprobar que las organizaciones se estructuran por medio de Unidades Organizativas que contienen agentes que llevan a cabo tareas específicas y diferenciadas.

Como se ha dicho, este capítulo describirá los cambios que se han aplicado al metamodelo. Estos cambios se han llevado a cabo por diversas razones:

- Integrar el concepto de artefacto, presentado en la sección 2.4, cuya funcionalidad podrá ser de gran utilidad en nuestro modelo.
- Actualizar el modelo para incorporar conceptos e ideas extraídas de otros OML propuestos recientemente.
- Incorporar conceptos requeridos por la plataforma THOMAS [GJR<sup>+</sup>09] que no se representaban en el VOM original, puesto que el objetivo del VOM es modelar sistemas que sean capaces de ser ejecutados en THOMAS.

Por otra parte, en la figura 4.2 se pueden apreciar de forma global las modificaciones que se han realizado al metamodelo tras introducir estos cambios.

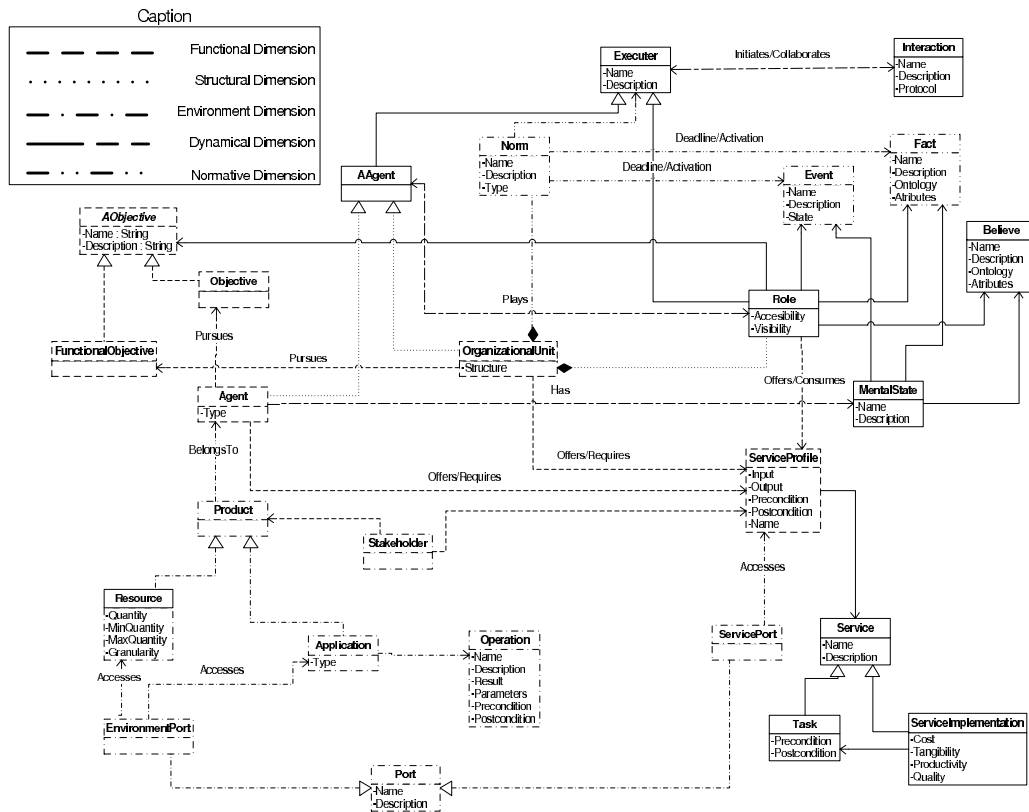


Figura 4.1: Metamodelo VOM original

Las siguientes subsecciones describirán de manera detallada las diferentes dimensiones del metamodelo, destacando los cambios y ampliaciones introducidas con respecto al modelo original.

## 4.1. Dimensión Estructural

La **Dimensión Estructural** (Figura 4.3) describe los componentes del sistema y sus relaciones. Permite definir los componentes estáticos de una organización, es decir, todos los elementos que son independientes de las entidades de tiempo de ejecución. Específicamente, define:

- la organización descrita por el sistema (*Organization*), que está compuesta de entidades atómicas (*Agent*) o de grupos de miembros de la organización (*Organizational Unit*), vistos como una única entidad desde el exterior. Esta organización existe como la agregación de los distintos grupos que la componen. El concepto de organización como elemento del modelo se ha introducido en esta modificación del metamodelo.

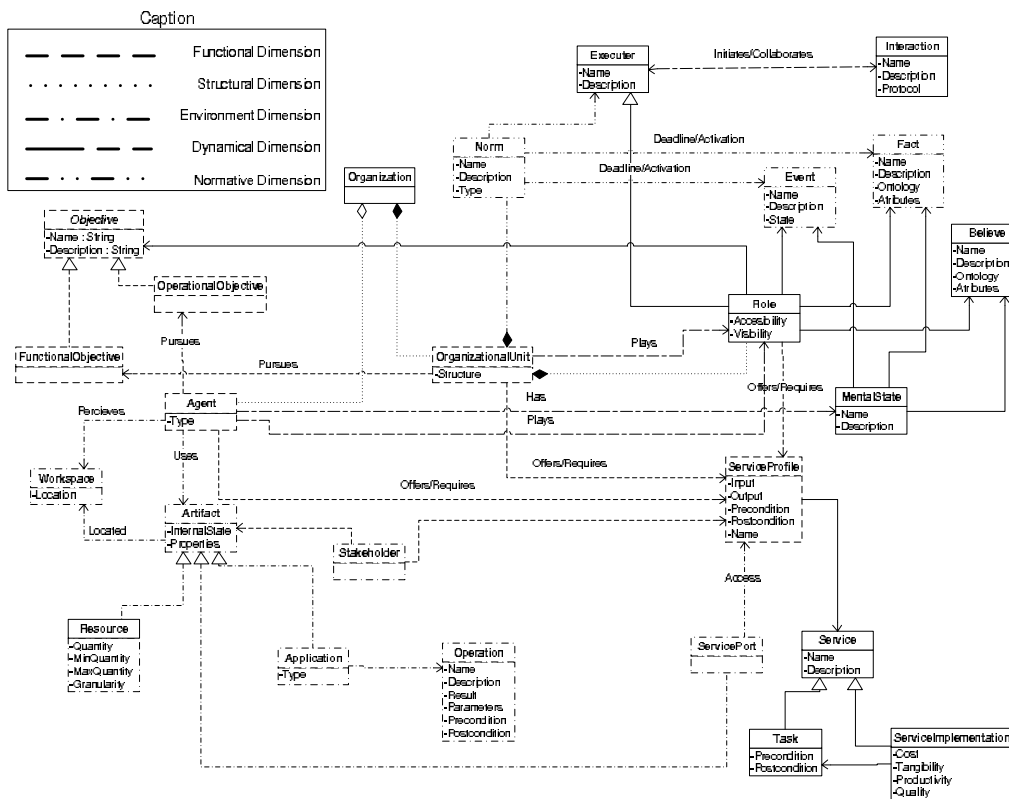


Figura 4.2: Extensión metamodelo VOM revisado

- las *Unidades Organizativas* (OUs) (entidad *Organizational Units*) del sistema, que pueden también incluir otras unidades de forma recursiva, además de agentes individuales.
- los *roles* (entidad *Role*) definidos dentro de las OUs. En la relación *contains* se puede especificar una cantidad mínima y máxima de entidades que pueden adquirir un rol. Para cada rol se han añadido dos atributos: *Accessibility* y *Visibility*. El atributo *Accessibility* indica si un rol puede ser adoptado por una entidad bajo demanda (externo) o siempre está predefinido por el diseño (interno). El atributo *Visibility* indica si las entidades pueden obtener información de este rol bajo demanda, desde el exterior de la unidad organizativa (rol público) o desde el interior, una vez que son miembros de esta unidad organizativa (es decir, roles privados). Se puede definir una jerarquía de roles con la relación *InheritanceOf*.
- las *relaciones sociales* de la organización (*SocialRelationships*). El tipo de una relación social entre dos entidades está relacionada con su

posición en la estructura de la organización (es decir, información, monitorización y supervisión). Por otra parte, se puede establecer una condición sobre cuándo está activa esta relación social gracias a los atributos para activar y desactivar que se han añadido a las relaciones sociales.

- las *normas* (entidad *norms*) que controlan el comportamiento global de los miembros de la OU.
- Todas las relaciones *contains* incluyen condiciones para activar un registro/desregistro de los elementos de una OU a través de su ciclo de vida.

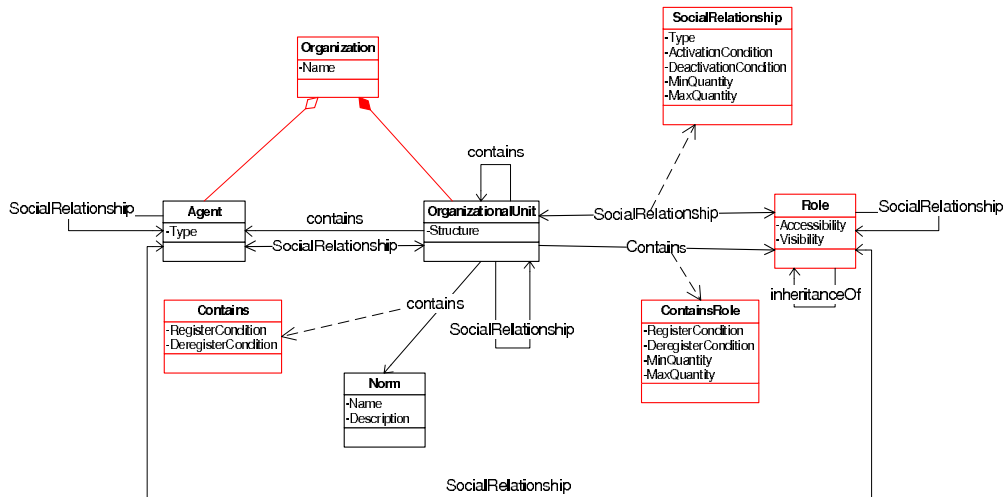


Figura 4.3: Diagrama de la *Dimensión Estructural*

En la *Dimensión Estructural* se ha suprimido con respecto al modelo originalmente presentado en [Arg08] la entidad AAgente y se ha introducido el concepto de Organización, que definimos que está formada por OUs y agentes. Además, en las relaciones *contains* se han añadido condiciones para el registro y desregistro de entidades de la OU, en este caso normas y roles. Esta característica proporciona una mayor flexibilidad en la estructura de la organización, ya que, por ejemplo, seremos capaces de desactivar normas o roles de nuestro sistema sin eliminarlas del modelo. Finalmente, se han añadido unas condiciones similares para activar o desactivar las relaciones sociales entre roles, OUs y agentes.

## 4.2. Dimensión Funcional

La **Dimensión Funcional** (Figura 4.4) detalla la funcionalidad específica del sistema, basada en servicios, tareas y metas, así como las interacciones del sistema, activadas por medio de metas o uso del servicio. Permite definir la funcionalidad de las unidades organizativas, los roles y los agentes del SMA. Específicamente, define:

- La funcionalidad de las Unidades Organizativas:
  - los *objetivos funcionales* (entidad *functional objectives*) que persigue la organización, es decir, requisitos no funcionales (softgoals) que pueden ser definidos para describir el comportamiento global de la organización.
  - los *grupos de interés* (entidad *stakeholders*) que interactúan con la OU.
  - los *servicios* que ofrece la organización (entidad *services*, que se describen usando perfiles de servicio). Una implementación específica de un servicio se define en la relación *offers* (atributo *ServiceImplementation*), que puede ser registrada en un directorio de servicios (atributo *RegisterPort*), con lo que otras entidades podrán encontrarlo. Las condiciones para controlar este proceso de registro también pueden ser especificadas (atributos *RegisterCondition* y *DeregisterCondition*). Estas condiciones han sido añadidas al metamodelo.
  - los *artefactos* (entidad *artifacts*) que la organización usa (relación *Uses*) para alcanzar sus objetivos, y que han sido introducidos en esta modificación para sustituir a los productos.
  - los servicios requeridos por la organización (relación *Requires*). Esta relación es similar a los anuncios de ofertas de trabajo de las organizaciones humanas, en el sentido que representa una necesidad de encontrar agentes capaces de proporcionar estos servicios como miembros de la organización.
  - las necesidades de la organización desde sus proveedores (relación *Requires*).
- la composición de metas:
  - Los componentes *Objective*, que pueden ser objetivos funcionales (entidad *FunctionalObjective*) (es decir, softgoals o requisitos no funcionales) u objetivos operacionales (hardgoals, entidad *OperationalObjective*). La nomenclatura de estos elementos ha cambiado, siendo el *AObjective* conocido como *Objective* y el antiguo *Objective* ahora es un *OperationalObjective*.

- Los *objetivos funcionales* representan los resultados esperados de las unidades organizativas, que se dividen en resultados específicos y medibles que sus miembros esperan conseguir.

- Los *objetivos operacionales* (entidad *OperationalObjective*) representan las metas operacionales, es decir, las metas específicas que deben cumplir los agentes o roles. También pueden ser refinadas en objetivos más específicos. Pueden estar relacionadas con una tarea o una interacción necesaria para satisfacer este objetivo.

- la funcionalidad de los roles:
  - las *metas* (*Objective*) perseguidas por un rol, que pueden ser *objetivos funcionales* (es decir, softgoal o requisitos no funcionales) u objetivos operacionales (es decir, hardgoals). En la relación *pursues*, se pueden definir las condiciones de activación o de fin de plazo para establecer una línea temporal en la cual se persigue el objetivo. En caso de una meta operacional (*OperationalObjective*), se puede definir una condición de satisfacción o fallo para establecer cuándo se cumple un objetivo, además de definir una tarea o interacción que permita alcanzar esta meta.
  - los *servicios* (*ServiceProfile*) relacionados con el rol, es decir, los servicios que el rol puede ofrecer o proporcionar a otras entidades.
  - las *tareas* (entidad *Task*) de las que el rol es responsable, es decir, la funcionalidad específica que se espera que el rol lleve a cabo.
- la funcionalidad de los agentes:
  - los *objetivos* (entidad *objective*) perseguidos por los agentes. Es posible definir condiciones de activación o de fin de plazo para establecer una línea de tiempo en la que se siga el objetivo. Además, una condición de fallo o satisfacción puede ser definida para establecer cuándo se cumple este objetivo.
  - los *servicios* (*ServiceProfile*) relacionados con el agente, es decir, los servicios que el agente puede ofrecer a otras entidades. Cuando se adopta un rol como miembro de una organización, el conjunto concreto de servicios que el agente podrá proporcionar se determina por su propio conjunto de servicios ofrecidos y aquellos relacionados con el rol adoptado.
  - las *tareas* (relación *Task*) de las que el agente es responsable, es decir, el conjunto de tareas que el agente es capaz de llevar a cabo.
- la composición de tareas y servicios:
  - El componente *Service* (antes conocido como *ATask*) describe la funcionalidad del servicio y representa tanto tareas, flujo de tareas y composición de servicios (con la relación *Invokes*). Este componente

*Service* puede descomponerse en otros componentes del mismo tipo, permitiendo el refinamiento de servicios o la composición de tareas.

- Una *tarea* (entidad *Task*) representa una funcionalidad básica, que consume y produce cambios en el estado mental de los agentes.

- La *interfaz de servicio* (*ServiceProfile*), que indica las condiciones de activación del servicio (precondiciones), sus parámetros de entrada y salida y sus efectos sobre el entorno (postcondiciones).

- La *funcionalidad* específica del servicio (*ServiceImplementation*), que describe una implementación concreta de un perfil de servicio.

- La composición de servicios, por medio de *invocaciones* (relación *Invokes*) entre servicios.

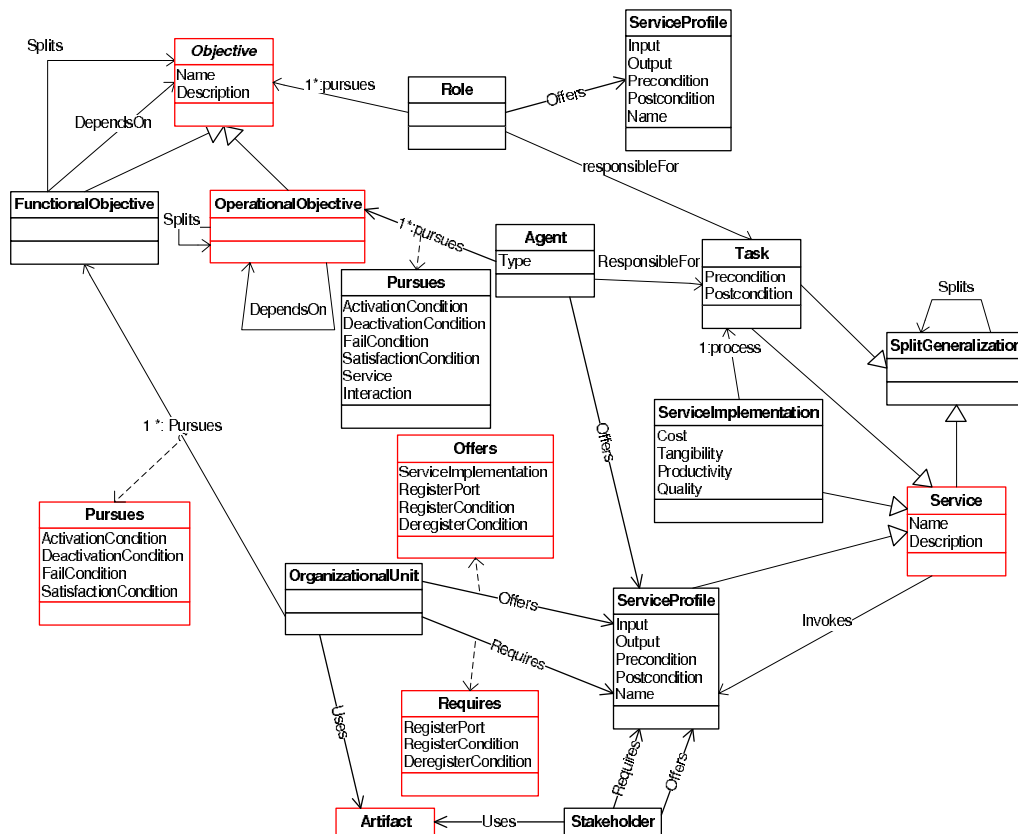


Figura 4.4: Diagrama de la Dimensión Funcional.

Como se ha comentado, se han hecho diversas aportaciones al modelo de la Dimensión Funcional. Por ejemplo, la entidad producto ha pasado a ser un artefacto y ya no se relaciona con OUs y grupos de interés por medio de relaciones *ofrece* y *requiere*, sino que lo hace por medio de una única

relación *usa*, que es la forma en que las entidades del sistema pueden hacer uso de los artefactos. La relación *consume* ha sido suprimida, ya que se ha considerado que emplear únicamente la relación *requiere* sería suficiente para expresar que se solicita un servicio. Como en la Dimensión Estructural, se han añadido distintas condiciones en la relación *persigue* entre un agente o OU y un objetivo. Estas condiciones son cuatro: de activación, de desactivación, de fallo y de satisfacción. Las dos primeras condiciones sirven para establecer el momento de perseguir ese objetivo y las dos últimas establecen qué hacer en caso de fallo o consecución de ese objetivo. Por último, las relaciones *ofrece* y *requiere* entre una OU y un perfil de servicio también poseen nuevos atributos. Estas relaciones indican el puerto de servicio en el que se registrará el servicio, proporcionando además las condiciones que deben cumplirse para registrar y desregistrar un servicio. La relación *ofrece* indica además la implementación concreta de ese perfil de servicio. Además, se ha modificado el concepto de producto por el de artefacto (este cambio se explicará más tarde al analizar la Dimensión de Entorno).

### 4.3. Dimensión Dinámica

La **Dimensión Dinámica** (Figuras 4.5 y 4.6) define el proceso de promulgación de roles, las interacciones entre agentes, además de los estados mentales de las entidades del sistema. Específicamente, define:

- los *roles* que una unidad organizativa puede jugar dentro de otras unidades organizativas (relación *Plays*), cuando se considera una única entidad. Los atributos *ActivationCondition* y *LeaveCondition* (añadidos en esta revisión) de esta relación indican en qué situación una OU adquiere o deja un rol, respectivamente.
- los *roles* jugados por cada agente. Los atributos *ActivationCondition* y *LeaveCondition* (introducidos en esta versión) de la relación *Plays* indican en qué situación un agente puede adquirir o dejar un rol.
- los *estados mentales* (entidad *MentalState*) del agente, usando creencias, eventos y hechos.
- la secuencia de interacciones:
  - Los *participantes* de la interacción (*Executer*). Las relaciones *Initiates* y *Collaborates* indican la secuencia de actividades (tareas y servicios) que han sido ejecutadas en una interacción. La relación *Collaborates* representa una actividad de respuesta. Esta entidad también ha sido introducida en esta revisión.
  - Las *performativas* (*InteractionUnit*) empleadas durante la interacción, añadidas en esta iteración del modelo.



- La entidad *Condition* permite cambiar la secuencia de interacciones dependiendo de una condición.

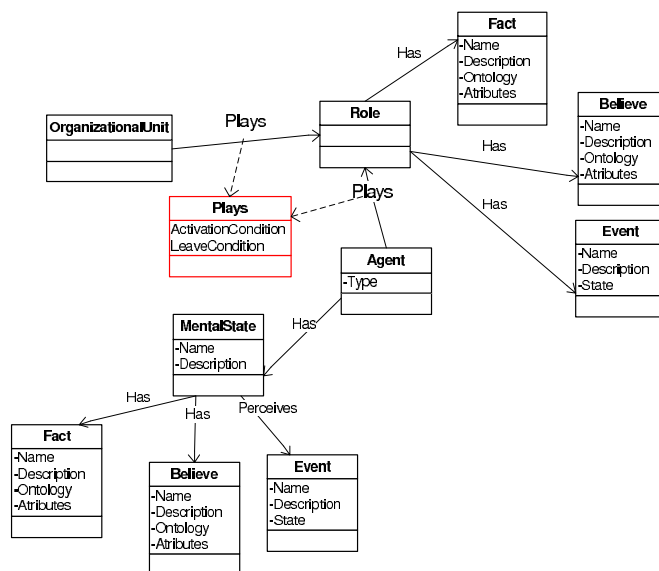


Figura 4.5: Diagrama de la Dimensión Dinámica (I).

Como en las dimensiones comentadas anteriormente, la Dimensión Dinámica también añade condiciones, en este caso, añade las condiciones en las que un agente o una OU puede tomar o dejar un rol. También se ha añadido la condición por la que una interacción puede cambiar su secuencia de unidades de interacción. En esta dimensión se agrupan además las entidades que pueden participar en una interacción bajo la entidad *participante*, y se añaden también las performativas que forman parte de una interacción por medio de la *unidad de interacción*.

## 4.4. Dimensión del Entorno

La **Dimensión del Entorno** (Figura 4.7) describe los elementos del entorno (artefectos), además de definir los espacios de trabajo (workspaces), que compondrán la totalidad del entorno. De manera más específica, define:

- Los *artefectos* (entidad *Artifact*) usados por los agentes (relación *Uses*) para conseguir sus objetivos. Estos artefactos se han integrado en esta dimensión. Los artefactos se pueden especializar en uno de los siguientes tipos (también se permite la definición de nuevos tipos de artefactos):

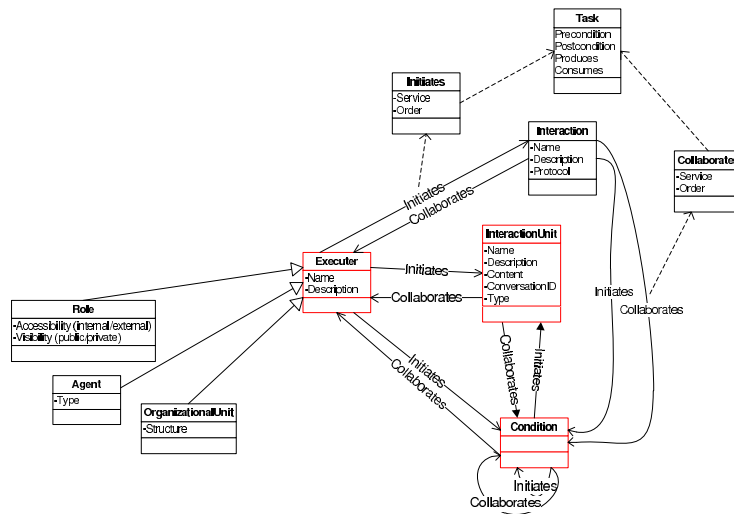


Figura 4.6: Diagrama de la Dimensión Dinámica (II).

– Los *recursos* (entidad *Resources*) representan objetos del entorno que no proporcionan una funcionalidad específica, pero resultan indispensables para la ejecución de tareas. Pueden ser o no consumibles, tener un estado inicial (*Quantity*), un umbral inferior y superior (atributos *MinQuantity*, *MaxQuantity*) y una granularidad (*granularity*).

– Las *aplicaciones* (entidad *Applications*) representan interfaces funcionales que se describen con un nombre (*name*), varios parámetros (*parameters*), precondiciones (*preconditions*), postcondiciones (*postconditions*) y resultados (*results*).

– Un *puerto de servicio* (entidad *ServicePort*) se considera como un servicio de publicación que ofrece la posibilidad de registrar y buscar servicios por su perfil.

– El *artefacto informativo* (entidad *InformativeArtifact*) es un mecanismo que devuelve información sobre acciones a un agente, dada una descripción parcial de su estado interno y teniendo en cuenta la vista parcial del entorno que tiene el artefacto.

– El *artefacto incentivo* (entidad *IncentiveArtifact*) es un mecanismo capaz de producir cambios en el entorno del sistema desde una perspectiva global, modificando las recompensas y sanciones que están activas en el sistema.

– El *artefacto coactivo* (entidad *CoerciveArtifact*) es un mecanismo centrado en producir modificaciones en el sistema produciendo cambios en las funciones de capacidad de los agentes, dada una descripción, posiblemente parcial, del SMA.

- El *entorno* del sistema (entidad *Environment*), formado por la unión de todos los *espacios de trabajo* (entidad *Workspace*), donde se encuentran ubicados los artefactos, y que pueden ser percibidos por los agentes. Estos espacios de trabajo pueden intersectarse o estar anidados (relaciones *Intersection* y *nesting*). Además, en este entorno se localiza la organización (entidad *Organization*). Estos conceptos también son novedosos de esta versión del modelo.

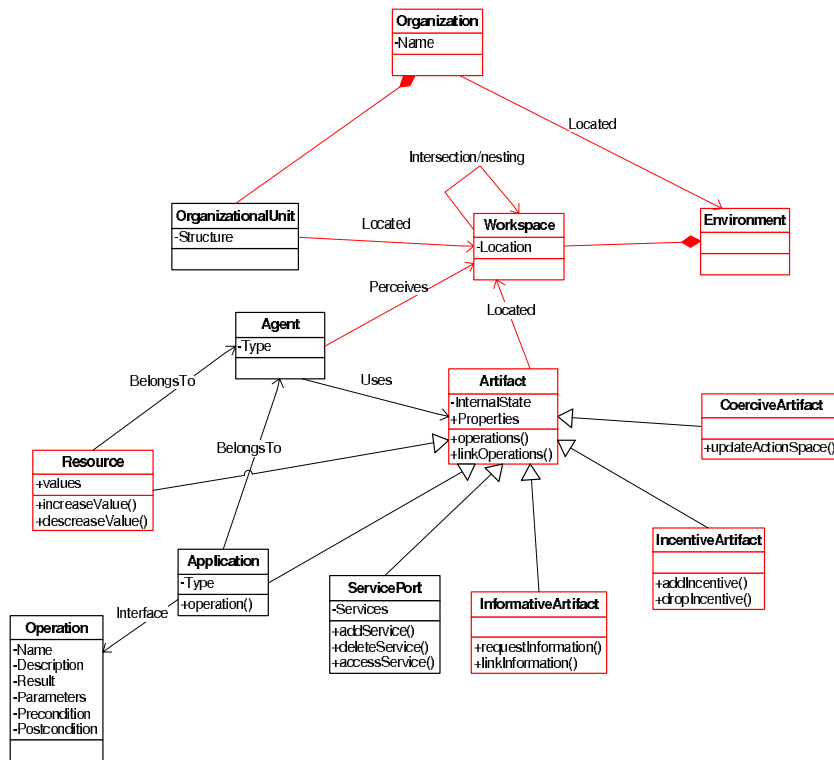


Figura 4.7: Diagrama de la Dimensión de Entorno

Esta dimensión es la que más cambios ha sufrido con respecto a la presentada originalmente. Ha sido rediseñada casi por completo para dar cabida al concepto de artefacto. Este concepto ha hecho que en la nueva versión del VOM los recursos y aplicaciones del sistema sean considerados como artefactos. También se han introducido los artefactos para mecanismos de organización. La entidad puerto ha sido suprimida debido a que su funcionalidad la aportan los propios artefactos, con lo que el único puerto que queda en el modelo, y además heredando de artefacto, es el puerto de servicio, que actuará como una pizarra para publicar los servicios. Además, cada entidad que pueda ser añadida al metamodelo y que encapsule una funcionalidad que

ayude a los agentes a lograr sus objetivos se modelará mediante un artefacto, debiendo heredar las características de éste. El otro concepto relevante introducido en el VOM es el *workspace* (o espacio de trabajo), usado como una forma de localización adecuada de las entidades, como los artefactos, dentro del metamodelo. La unión de todos los workspaces dará lugar al entorno del sistema, donde se localizará la organización que se pretende diseñar.

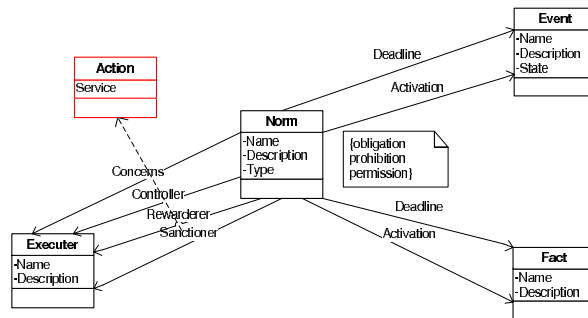
Los artefactos han hecho que los puertos (entidad *Port*) no sean necesarios en el metamodelo. Los puertos del entorno eran los encargados de controlar la entrada y salida a elementos del entorno como recursos y aplicaciones. Con la introducción de los artefactos, este tipo de entidades deja de tener utilidad, puesto que son los propios artefactos los que encapsularán el control de acceso a los productos. El único puerto que queda en el metamodelo es el puerto de servicio, que seguirá siendo un punto de publicación de servicios que se encuentran en el entorno del sistema.

## 4.5. Dimensión Normativa

Por último, la **Dimensión Normativa** (Figura 4.8) describe las restricciones normativas sobre el comportamiento de las entidades del sistema, incluyendo normas organizativas y metas normativas que los agentes deben seguir, incluyendo sanciones y recompensas. Específicamente, define:

- el concepto de *norma* (entidad *Norm*), que representa una regulación específica. Las propiedades de la norma detallan todos los hechos y eventos del entorno que provocan la activación o desactivación de la norma.
- La entidades *Executer* sobre la que se ejecuta la norma (relación *Concerns*).
- El *participante* (*Executer*) que es responsable de: monitorizar el cumplimiento de la norma (relación *Controller*); aplicar castigos (relación *Defender*); y/o aplicar recompensas (relación *Rewarderer*).
- El atributo *Service* de todas estas relaciones indica qué tarea o servicio será invocado cuando se monitorice esta norma o cuando se castigue o se premie a los agentes.

En esta dimensión se ha producido un cambio únicamente de nomenclatura. Se ha modificado el nombre del atributo *ATask* de la entidad *Action* por el de *Service*, para no perder la consistencia con el resto de dimensiones del metamodelo. Se ha considerado que el resto de entidades de esta dimensión pueden representar de manera óptima las normas de un SMA.

Figura 4.8: *Diagrama de la Dimensión Normativa*

## 4.6. Discusión

Este metamodelo nos presenta de forma gráfica una forma de definir organizaciones virtuales desde cinco puntos de vista aportados por las cinco dimensiones organizativas descritas: Estructural, Funcional, Dinámica, de Entorno y Normativa. Cada una de estas dimensiones posee un modelo en el que se encuentran definidos sus elementos característicos. Estos modelos han sido modificados para mejorar la capacidad de representación de este metamodelo, prestando especial atención al modelado del entorno, en el que se han introducido conceptos de la propuesta Agents & Artifacts [RVO07], en concreto el artefacto y el espacio de trabajo, mejorando así un elemento esencial en un sistema multiagente.

Cabe destacar que parte de este trabajo de actualización se ha realizado en colaboración con E. García y A. Giret, responsables del desarrollo de una herramienta CASE, llamada emfGORMAS [GAG09], que permite el diseño gráfico de estos metamodelos.

Además, el metamodelo ha sido actualizado incluyendo nuevas funcionalidades que serán útiles tanto en la fase de diseño de un SMA, al realizar los metamodelos, como en la fase de implementación y de puesta en marcha, incluyendo conceptos requeridos por la arquitectura THOMAS [CGJ<sup>+</sup>09, GJR<sup>+</sup>09], centrada en el desarrollo de organizaciones virtuales siguiendo una aproximación orientada a servicios y en la que se ejecutarán los sistemas diseñados mediante VOM.



# Capítulo 5

## Formalización de una Organización Virtual

Como se comentó en el capítulo 2, el modelado de OCMAS se puede realizar empleando metamodelos o bien mediante aproximaciones formales, basadas por ejemplo en modelos de Kripke.

En este capítulo definiremos el concepto de Organización Virtual (VO) de manera formal, teniendo en cuenta sus dimensiones organizativas. Esta formalización se centrará en tres elementos de la VO: (i) la Especificación Organizativa (*Organizational Specification, OS*), que detalla el conjunto de elementos estáticos de la organización; (ii) la Entidad Organizativa (*Organizational Entity, OE*), que representa la instanciación de los elementos de la *OS*; y (iii) la Dinámica Organizativa (*Organizational Dynamics,  $\phi$* ), que relaciona los elementos de la *OS* con elementos de la *OE*.

**Definición 9.** Una organización virtual se define, en un espacio de tiempo  $t$ , como una tupla  $O^t = \langle OS^t, OE^t, \phi^t \rangle$  donde:

- *OS se refiere a la Especificación Organizativa. Se define como  $OS = \langle SD, FD, ED, ND \rangle$  donde:*
  - *SD es la Dimensión Estructural.*
  - *FD es la Dimensión Funcional.*
  - *ED es la Dimensión del Entorno.*
  - *ND es la Dimensión Normativa.*
- *OE se refiere a la Entidad Organizativa, que representa a los elementos dinámicos del sistema.*
- *$\phi$  permite relacionar OS con OE.*

La Organización Virtual cambiará durante el tiempo modificando sus estados. Estos cambios ocurrirán después de un cambio en el entorno, y la VO se moverá de un estado a otro usando una transición. Las siguientes subsecciones describirán con detalle estos tres elementos que definen formalmente una Organización Virtual.

## 5.1. Especificación Organizativa

La Especificación Organizativa detalla el conjunto de elementos 'estáticos' de la organización, conteniendo unidades organizativas, roles, normas y el resto de elementos que forman las dimensiones de una Organización Virtual. La Estructura Organizativa se compone de (i) la *Dimensión Estructural*, que contiene roles, unidades organizativas y sus relaciones; (ii) la *Dimensión Funcional*, describiendo objetivos, funciones y servicios de la organización; (iii) la *Dimensión del Entorno*, que describe los artefactos y espacios de trabajo del entorno de la organización; y (iv) la *Dimensión Normativa*, que define las normas que controlan el sistema.

### 5.1.1. Dimensión Estructural

La *Dimensión Estructural* describe los componentes del sistema y sus relaciones. Permite definir los componentes estáticos de la organización, es decir, todos los elementos que son independientes de las entidades que finalmente se ejecutan. De forma más específica, define las unidades organizativas y los elementos estructurales que contiene: roles y las relaciones entre roles.

**Definición 10.** *La Dimensión Estructural (Structural Dimension, SD) de una Organización Virtual se define como  $SD = \langle R, OU, Relations \rangle$  donde:*

- *R se refiere a los roles de la organización.*
- *OU es el conjunto de unidades organizativas.*
- *Relations es un conjunto de relaciones, definidas como  $Relations = \langle SocialRelations, StructRelations, DimensionRelations \rangle$  donde:*
  - *SocialRelations se refiere a las Relaciones Sociales entre roles, que pueden formalizarse como:*

$$SocialRelations = \begin{cases} inf : & R \rightarrow R \\ col : & R \rightarrow R \\ sup : & R \rightarrow R \\ comp : & R \rightarrow R \end{cases}$$



donde: *inf* se refiere a las relaciones de información, que muestra cómo se establecen enlaces de información sobre el estado actual de la organización; *col* (colaboración) permite monitorizar las actividades de los agentes y lanzar respuestas apropiadas si ocurre un fallo o una violación, permitiendo colaboración entre agentes; *sup* (supervisión) define que un agente puede transferir o delegar uno o más de sus objetivos a un agente subordinado, que está obligado a alcanzar estos objetivos; y *comp* (compatibilidad) refleja que un agente tomando un rol puede tomar a la vez otro rol en la organización.

- *StructRelations* se refiere a las relaciones estructurales que define la estructura de la organización, que se puede formalizar como:

$$\text{StructRelations} = \begin{cases} \text{RoleHier} : R \rightarrow R \\ \text{Contains} : OU \rightarrow 2^{OU} \\ \text{Roles} : OU \rightarrow 2^R \end{cases}$$

donde: *RoleHier* representa la jerarquía entre los roles de la organización; *Contains* define la topología de la organización mediante relaciones entre unidades organizativas; y *Roles* define los roles que se encuentran dentro de una unidad organizativa.

- *DimensionRelations* permite relacionar esta dimensión con otras, a través del elemento *OU*, y se puede formalizar como:

$$\text{DimensionRelations} = \begin{cases} \text{Norms} : OU \rightarrow 2^N \\ \text{Services} : OU \rightarrow 2^S \\ \text{Goals} : OU \rightarrow 2^G \\ \text{Workspaces} : OU \rightarrow 2^{WS} \end{cases}$$

donde: *Norms* define las normas, descritas en la dimensión normativa, que gobiernan una unidad organizativa; *Services* relaciona una unidad organizativa con los servicios que contiene; *Goals* describe los objetivos que se necesitan alcanzar dentro de una unidad organizativa; y *Workspaces* detalla los espacios de trabajo en los que una unidad organizativa puede ubicarse.

**Propiedades de las relaciones** A continuación de definen las propiedades que presentan las relaciones de la Dimensión Estructural:

La relación social *inf* es simétrica, un rol puede proporcionar información a un segundo rol, y viceversa. Adicionalmente, esta relación presenta la propiedad transitiva, ya que los agentes pueden construir una cadena de

información, y la propiedad reflexiva, ya que un agente puede enviar información a sí mismo. Las relaciones *col* y *sup* son ambas asimétricas, ya que un agente no puede monitorizar o supervisar al agente que le monitoriza o supervisa; reflexivas, porque un agente puede colaborar consigo mismo o supervisarse; y transitivas, permitiendo crear una cadena de mando dentro de la organización.

La relación de compatibilidad (*comp*) posee las propiedades reflexiva y transitiva, porque un rol es compatible consigo mismo y un rol es compatible con los roles que tienen una relación de compatibilidad con sus roles compatibles. Es interesante destacar que la relación *comp* no es simétrica (por ejemplo,  $comp(r_1, r_2)$  no siempre implica  $comp(r_2, r_1)$ ). Por ejemplo, la relación  $comp(Professor, Teacher)$  es correcta, porque un catedrático puede actuar como un profesor normal en cualquier momento, pero un profesor no está capacitado para tomar el rol de catedrático a no ser que posea la acreditación requerida. Finalmente, las relaciones *RoleHier* y *comp* están relacionadas, ya que un agente jugando un rol especializado es capaz de jugar su rol generalizado. Formalmente:

$$\forall r_1, r_2 \in R : RoleHier(r_1, r_2) \rightarrow comp(r_2, r_1) \quad (1)$$

Sean  $r_1, r_2 \in R$  dos roles pertenecientes a *OS*. Las relaciones de información, colaboración y supervisión definen las siguientes relaciones de forma implícita:

$$sup(r_1, r_2) \rightarrow col(r_2, r_1) \quad (2)$$

$$col(r_1, r_2) \rightarrow inf(r_1, r_2) \wedge comp(r_2, r_1) \quad (3)$$

Esto significa que una relación de supervisión entre dos agentes implica que el agente supervisado colaborará con el agente supervisor a alcanzar sus objetivos. Además, una relación de colaboración entre dos roles implicará que exista un enlace de información entre ellos y que el segundo rol de la relación sea compatible con el primero.

La relación *Contains* del conjunto *StructRelations* tiene las siguientes propiedades: (i) asimétrica, ya que una OU no puede estar contenida en una OU que la contiene; (ii) transitiva, porque se considera que una OU contenida dentro de otra OU también está contenida dentro de los predecesores de la OU que la contienen; y (iii) irreflexiva ya que una OU no puede contenerse en sí misma. De manera similar, la relación *RoleHier* tiene las mismas propiedades de la relación *Contains*, porque un rol no puede tener una relación de herencia consigo mismo, las relaciones entre roles son transitivas para permitir definir una jerarquía de roles completa y un rol subordinado no puede ser el supervisor de su supervisor.

**Propiedades de las entidades** Después de formalizar las propiedades que tienen las relaciones de la dimensión estructural, es necesario definir las propiedades presentadas por las entidades que pueblan esta dimensión. La primera relación de las entidades de esta dimensión establece que un agente localizado en una unidad organizativa que está contenida dentro de otra OU puede jugar los roles de su OU y la OU que lo contiene. Formalmente:

$$\forall OU_1, OU_2 \in OU : \text{Contains}(OU_1, OU_2) | \forall r_1 \in \text{Roles}(OU_1) \\ \wedge \forall r_2 \in \text{Roles}(OU_2) \rightarrow \text{comp}(r_2, r_1)$$

Cabe destacar que la relación *Roles* es recursiva. Los roles que ofrece una OU no son sólo sus propios roles, sino que además son los roles de sus OUs predecesoras. Formalmente:

$$\forall o \in OU : \forall r \in \text{Roles}(o) \rightarrow r \in \text{Roles}(o) \vee r \in \text{Roles}(o_1) : o \in \text{Contains}(o_1)$$

**Propiedades de las OU** Las relaciones entre unidades organizativas permiten que se puedan definir tres tipos distintos de estructura de una organización:

- *'hierarchy'*. Define una jerarquía dentro de una unidad organizativa (OU) usando agentes que toman el rol de supervisor, que está a cargo de sus agentes subordinados, creando relaciones de supervisión. Formalmente,  $\exists r \in \text{Roles}(OU) : \forall r_i \neq r \in \text{Roles}(OU) \rightarrow \text{sup}(r, r)$ . Si un diseñador quiere hacer que su sistema sea más estricto, puede prohibir la comunicación entre agentes subordinados.
- *'team'*. En este tipo de relación, todos los agentes tienen relaciones de coordinación entre ellos. Formalmente, se puede definir como  $\forall r_1, r_2 \in \text{Roles}(OU) : \text{col}(r_1, r_2)$ .
- *'plain'*. Este tipo no define una topología concreta, pero establece relaciones de información entre roles. Formalmente,  $\exists r_1, r_2 \in \text{Roles}(OU) : \text{inf}(r_1, r_2)$ .

### 5.1.2. Dimensión Funcional

La *Dimensión Funcional* detalla la funcionalidad específica del sistema, basada en servicios, tareas y objetivos, además de las interacciones del sistema, activadas por medio de objetivos o utilización de servicios. Permite definir la funcionalidad de las unidades organizativas, roles y agentes del SMA, incluyendo los servicios y objetivos que estas entidades ofrecen o consumen.

**Definición 11.** *La Dimensión Funcional (FD) de la Estructura Organizativa en una Organización Virtual se define como  $FD = \langle G, S, Ta, FuncRel \rangle$  donde:*

- $G$  representa los objetivos que persigue la organización.
- $S$  es el conjunto de servicios que el sistema ofrece o requiere.
- $Ta$  son las tareas que componen los servicios.
- $FuncRel = \langle GT, Client, Provider, Pursues, Task, Invoke \rangle$  es el conjunto de relaciones de esta dimensión, donde:
  - $GT : G \rightarrow 2^G$  es el árbol de objetivos (Goal Tree) de la organización, que describe las dependencias entre los diferentes objetivos de la organización.
  - $Client : S \rightarrow 2^R$  relaciona un servicio con el conjunto de roles que lo usan.
  - $Provider : S \rightarrow 2^R$  relaciona un servicio con el conjunto de roles que lo ofrecen.
  - $Obtains : S \rightarrow 2^O$  describe el conjunto de objetivos perseguidos por los servicios, definiendo así la funcionalidad del servicio.
  - $Achieves : Ta \rightarrow 2^O$  define el conjunto de objetivos que se consiguen al ejecutar una tarea.
  - $Task : S \rightarrow 2^{Ta}$  muestra la división de servicios en múltiples tareas.
  - $Invoke : S \rightarrow 2^S$  describe las dependencias entre servicios, mostrando los servicios que necesitan invocar a otros servicios para completar su funcionalidad, permitiendo así la composición de servicios.

**Propiedades de las relaciones** La relación Goal Tree tiene las propiedades irreflexiva, asimétrica y transitiva. Un objetivo no se puede relacionar consigo mismo ni con su objetivo predecesor. Un objetivo se relaciona con los sucesores de sus sucesores.

Se debe asegurar que el proveedor de un servicio debe ser un rol contenido en el mismo OU que el servicio. Formalmente:

$$\forall o \in OU \wedge \forall s \in Services(o) \rightarrow Provider(s) \subseteq Roles(o) \quad (4)$$

Esta restricción asegura que los servicios de una OU se proporcionarán dentro de ella únicamente, pero pueden ser accedidos por agentes de otras OUs (por ejemplo, usando la relación Invoke).

Como se ha indicado en esta sección, la relación Invoke permite a los servicios invocar otros servicios para alcanzar sus objetivos. Para ejecutar esta operación, se debe asegurar que el proveedor del servicio invocante sea cliente del servicio invocado. Formalmente,

$$\forall s_1, s_2 \in S, s_2 \in \text{Invokes}(s_1) : \forall r_1 \in \text{Provider}(s_1) \rightarrow \exists r_2 \in \text{Client}(s_2) \wedge r_1 = r_2$$

Un aspecto clave para el diseñador del sistema es asegurar que los servicios localizados en una unidad organizativa deben ayudar a alcanzar sus objetivos. Formalmente, se describe como:

$$\forall o \in OU, \forall s \in \text{Services}(o) \wedge \forall g_1 \in \text{Pursues}(s) : \exists g_2 \in \text{Goals}(o) \rightarrow g_2 \in GT(g_1)$$

También se debe obligar a un servicio a que cumpla todos sus objetivos. Si esto no es posible debido a que las tareas que forman ese servicio no alcanzan dicho objetivo, entonces un servicio debe invocar a otro servicio que incluya una tarea que sí logre alcanzar el objetivo deseado. Formalmente, lo expresamos así:

$$\begin{aligned} \forall g \in \text{Obtains}(s_1) \rightarrow (\exists t \in \text{Task}(s_1) \wedge g \in \text{Achieves}(t)) \\ \vee (\exists s_2 \in S \wedge g \in \text{Obtains}(s_2) \wedge s_2 \in \text{Invokes}(s_1)) \end{aligned}$$

### 5.1.3. Dimensión del Entorno

La Dimensión del Entorno describe los artefactos, entidades que pueblan el entorno de un SMA. Esta dimensión usa el concepto de artefacto [RVO07], un elemento introducido por el marco de trabajo Agents & Artifacts (A&A). Estos elementos son empleados por los agentes para ayudarles a alcanzar sus objetivos, ya que los artefactos no tienen objetivos asociados. Además, el marco de trabajo A&A presenta el concepto de workspace, usado para definir la topología del entorno de un SMA.

**Definición 12.** *La Dimensión del Entorno de una Organización Virtual se define como  $ED = \langle WS, AR, EnvFunc \rangle$  donde:*

- *WS es el conjunto de workspaces que componen el entorno de un SMA, donde un  $ws \in WS$  se define como  $ws = \langle Loc \rangle$  donde Loc se refiere a la localización del workspace dentro del entorno.*

- *AR* es el conjunto de artefactos, donde un artefacto  $ar \in AR$  se define  $ar = \langle PR, OP, LO, St \rangle$ , donde:
  - *PR* son las propiedades observables de un artefacto que los agentes pueden comprobar sin hacer operaciones sobre él.
  - *OP* es el conjunto de operaciones que los agentes pueden ejecutar cuando interactúan con ellos.
  - *LO* se refiere a las operaciones de enlace, que permite la composición de artefactos y los artefactos distribuidos uniendo artefactos.
  - *St* es el estado interno de un artefacto.
- *EnvFunc* =  $\langle Located, Composition \rangle$  es el conjunto de funciones que actúan sobre los elementos del entorno, donde:
  - *Located* :  $AR \rightarrow 2^{WS}$  describe el conjunto de workspaces en los que se encuentra un artefacto.
  - *Composition* :  $WS \rightarrow 2^{WS}$  permite definir relaciones de intersección y anidamiento entre los workspaces que forman el entorno.

**Propiedades de las relaciones** La relación *Composition* es reflexiva y simétrica, porque un workspace puede intersectarse consigo mismo y no importa la dirección de la relación.

Además, es necesario que un artefacto esté contenido como mínimo en un workspace. Formalmente, lo podemos expresar como:

$$\forall ar \in Ar : \exists ws \in WS \rightarrow ws \subseteq Located(ar) \quad (5)$$

#### 5.1.4. Dimensión Normativa

La Dimensión Normativa describe restricciones normativas sobre el comportamiento de las entidades del sistema, incluyendo sanciones y recompensas. Se basa en un trabajo previo presentado en [CAB10], donde se define un modelo normativo que describe las normas que controlan el sistema.

Antes de presentar la definición de Dimensión Normativa de una Organización Virtual, supongamos la existencia de un lenguaje de predicados de primer orden  $\mathcal{L}$  cuyo alfabeto incluye: los símbolos cuantificadores  $\{\forall, \exists\}$ ; los conectores lógicos  $\{\wedge, \vee, \rightarrow, \neg\}$ ; parentésis, corchetes y otros signos de puntuación; y un conjunto infinito de variables. Además, el alfabeto contiene predicados no-lógicos, constantes y símbolos de funciones. Asumiremos la definición estándar de *wffs* (fórmulas bien formadas). También se usa la noción estándar de sustitución de variables en una *wff*, donde  $\sigma$  es un conjunto

finito, posiblemente vacío, de pares  $X/y$  donde  $X$  es una variable en una *wff* e  $y$  es un término. Si la *wff* obtenida al aplicar una substitución no tiene variables, se definirá como bien fundada o parcialmente bien fundada.

**Definición 13.** *La Dimensión Normativa de una Organización Virtual se define como  $ND = \langle N, >_n \rangle$  donde:*

- $N$  es el conjunto de normas del sistema.
- $>_n$  es una relación de orden entre normas, definiendo la prioridad entre ellas. Esta relación establece una relación de orden total entre las normas que gobiernan el sistema, evitando la confusión de la prioridad cuando se ejecuta una norma.

Formalmente, una norma se define como:

**Definición 14.** *Una norma  $n \in N$  se define como  $n = \langle D, CO, AC, EX, SA, RE \rangle$  donde:*

- $D = \{O, F\}$ , es el operador deóntico. Este trabajo considera únicamente obligaciones ( $O$ ) y prohibiciones ( $F$ ) que imponen restricciones en el comportamiento de los agentes; mientras que los permisos no han sido considerados debido a que pueden ser definidos como operadores que invalidan la activación de obligaciones y prohibiciones.
- $CO$  es una fórmula lógica que representa la condición normativa o acción que puede llevarse a cabo en caso de normas de obligación, o debe ser evitada en caso de normas de prohibición.
- $AC, EX$  son *wffs* que determinan las condiciones de activación y expiración de normas, respectivamente.
- $SA, RE \in S$  son expresiones que describen las acciones (sanciones,  $SA$ ; y recompensas,  $RE$ ) que serán llevadas a cabo en caso de violación o cumplimiento de normas, respectivamente.

**Propiedades de las relaciones** La función de prioridad  $>_n$  es asimétrica y transitiva, definiendo una relación unívoca entre las normas que controlan el sistema.

La topología del sistema también definirá nuevas relaciones de orden entre normas. Si una OU llamada  $ou_2$  está contenida en una OU llamada  $ou_1$ , sus normas deben tener más prioridad que las normas de  $ou_1$ . Formalmente,

$$\forall ou_1, ou_2 : n_1 \in Norms(ou_1) \wedge n_2 \in Norms(ou_2) \\ \wedge Contains(ou_1, ou_2) \rightarrow n_2 >_n n_1$$

## 5.2. Entidad Organizativa

La Entidad Organizativa de una Organización Virtual es el conjunto de elementos activos de la organización. Estos elementos pueden cambiar a través del tiempo. Son considerados como los elementos dinámicos del sistema.

**Definición 15.** *La Entidad Organizativa de una Organización Virtual se define como  $OE = \langle A, GR, AN, AS \rangle$  donde:*

- *$A$  es el conjunto de agentes que puebla la Organización Virtual.*
- *$GR$  es el conjunto de grupos que se encuentran actualmente en el sistema. Un grupo es una instanciación de una Unidad Organizativa.*
- *$AN \subseteq N$  es el conjunto de normas activas en el sistema.*
- *$AS \subseteq S$  es el conjunto de servicios que la organización proporciona actualmente.*

Los agentes ( $A$ ) que están poblando el sistema están jugando un rol y localizados en un grupo ( $GR$ ), como se describirá en la siguiente sección. Una unidad organizativa define un patrón organizativo para los agentes que se encuentran en él, pero esto no define los agentes concretos que deberían poblarla. Por el contrario, un grupo es una instancia concreta de una unidad organizativa, definiendo un conjunto de agentes que lo pueblan. Un tipo de unidad organizativa puede ser instanciada por diferentes grupos.

**Propiedades de las relaciones** Para definir formalmente tanto  $AN$  como  $AS$ , es necesario usar dos funciones que se encontrarán en  $\phi$  y que se explicarán en la siguiente sección. La función  $activeNorm(n) = \{0, 1\}$  establece si una norma del  $ND \in OS$  se encuentra activa actualmente. De forma similar,  $providedService(s) = \{0, 1\}$  establece si un servicio se proporciona en el sistema en ese momento. Por tanto, ambos conjuntos se definen formalmente como:

$$\forall n \in AN \rightarrow n \in N \wedge activeNorm(n) = 1 \quad (6)$$

$$\forall s \in AS \rightarrow s \in S \wedge providedService(s) = 1 \quad (7)$$

## 5.3. Dinámica Organizativa

La Dinámica Organizativa representa las relaciones entre elementos de la Estructura Organizativa y la Entidad Organizativa.



**Definición 16.** *La Dinámica Organizativa de una Organización Virtual se define como  $\phi = \langle plays, inUnit, provides, perceives, isUnit, isAble, activeNorm, activeService \rangle$  donde:*

- *$plays : A \rightarrow 2^R$  es una función que relaciona un agente con el conjunto de roles que está jugando en la organización.*
- *$inUnit : A \rightarrow 2^{GR}$  es la función que describe los grupos donde se encuentra un agente.*
- *$provides : A \rightarrow 2^S$  representa el conjunto de servicios que proporciona un agente.*
- *$perceives : A \rightarrow 2^{WS}$  representa el conjunto de workspaces que un agente es capaz de percibir.*
- *$isUnit : GR \rightarrow OU$  define el tipo de unidad organizativa instanciada por un grupo.*
- *$isAble : A \times S \rightarrow \{0, 1\}$  define si un agente es capaz de usar un servicio.*
- *$activeNorm : N \rightarrow \{0, 1\}$  establece si una norma está actualmente activa en el sistema.*
- *$activeService : S \rightarrow \{0, 1\}$  muestra si un servicio está activo en el sistema.*
- *$compGoal : GR \times G \rightarrow \{0, 1\}$  define si un objetivo de un grupo se está cumpliendo en ese momento.*

**Propiedades de las relaciones** Las relaciones *plays*, *inUnit*, *provides* y *perceives* permiten a los agentes jugar distintos roles, estar localizados en diferentes grupos, proporcionar diferentes servicios y percibir distintos workspaces en la organización, respectivamente. La relación *isUnit* permite conocer el tipo de Organización Virtual que un grupo concreto está instanciando. Hay un grupo de tres funciones que permitirá conocer algunas propiedades del sistema. La primera de las funciones es *isAble*, que es una función muy interesante en caso de necesitar permitir a un agente concreto ejecutar un servicio incluso si el rol o los roles que toma no permite esta operación. Finalmente, si es necesario conocer las normas y servicios activos en el momento, las funciones *activeNorm* y *activeService* deben ser ejecutadas. Como se estableció en la sección 5.2, ambas funciones son necesarias para definir formalmente el conjunto de normas y servicios activos de la Entidad Organizativa del sistema.

Después de definir las relaciones de  $\phi$ , es posible definir formalmente la situación donde un agente juega un rol dentro de una unidad y un escenario en el que un agente está dentro de una unidad organizativa jugando un rol, como se puede comprobar en las siguientes ecuaciones. Cabe recordar que la función *Roles* es recursiva tal y como se explicó en la sección 5.1.1.

$$\forall r \in \text{plays}(a) \rightarrow \exists o \in OU \wedge \exists g \in \text{inUnit}(a) : \text{isUnit}(g) = o \wedge r \in \text{Roles}(o)$$

$$\forall g \in \text{inUnit}(a) \rightarrow \exists o \in OU \wedge \text{isUnit}(g) = o \wedge \exists r \in \text{Roles}(o) : r \in \text{plays}(a)$$

La primera ecuación establece que un agente sólo puede jugar los roles proporcionados por los grupos en los que se encuentra. Estos roles será los que proporcionan las unidades organizativas instanciadas por estos grupos. Por su parte, la segunda ecuación define que un agente debe jugar al menos un rol de cada uno de los grupos en los que se encuentra.

Otra propiedad interesante de las entidades de la dimensión estructural se relaciona con sus objetivos. Con el propósito de alcanzar sus objetivos, una OU tiene que asegurarse que todos los objetivos de sus OUs subordinadas se completen. Habitualmente, los objetivos de las OUs subordinadas son especializaciones de los objetivos de las OUs que los contienen. Formalmente,

$$\begin{aligned} \forall g_1 \in \text{Goals}(o_1) : \text{compGoal}(o_1, g_1) = 1 \rightarrow \forall o_2 \in \text{Contains}(o_1) \wedge \\ \forall g_2 \in \text{Goals}(o_2) \wedge \text{compGoal}(o_2, g_2) \end{aligned}$$

## 5.4. Sistema Multiagente basado en Organizaciones Virtuales

Sea  $VO$  el universo de todas las posibles organizaciones  $O$ . Un sistema multiagente basado en organizaciones virtuales es una estructura  $MAS = (VO, D)$  donde, para cada instante  $t \in T$ ,  $D^t \subseteq VO \times VO$  es la dinámica estructural del sistema, en el que en cada estado de la organización  $O \in VO$ , en un instante  $t \in T$ , existe un conjunto de posibles estados siguientes de la organización, denotados por  $D^t(O) \subseteq VO$ . Por tanto, para cada  $t \in T$  se cumple que  $O^{t+1} \in D^t(O^t)$ , con lo que una organización sólo cambiará a otro estado al que le esté permitido llegar desde el estado origen.

Para producir este cambio no es necesario que cambie toda la organización al completo, sino que se pueden producir cambios únicamente en uno

de los elementos que componen la organización, es decir, en los elementos  $OS, OE, \phi$ . Por tanto,

$$(OS^{t+1}, OE^{t+1}, \phi^{t+1}) \in D^t((OS^t, OE^t, \phi^t)) \leftrightarrow OS^{t+1} \in D_{OS}^t(OS^t) \\ \wedge OE^{t+1} \in D_{OE}^t(OE^t) \wedge \phi^{t+1} \in D_{\phi}^t(\phi^t)$$

## 5.5. Discusión

En la sección 2.3 del capítulo 2 se realizó un análisis de las propuestas de formalización más relevantes. El trabajo que hemos realizado en este capítulo está basado en muchas de ellas, en concreto en OperA [Dig03], MOISE+ [HSB02], SODA [MODR06] y PopOrg [dRCD08].

De este modo, ... (y recuerdas brevemente lo que proponían). Por otro lado, para modelar la estructura de una organización, la propuesta OperA.....

De este modo, nuestra propuesta de evolución temporal de la organización se basa principalmente en la propuesta de PopOrg [dRCD08] que modela la dinámica relacionada con la forma en que la estructura del sistema cambia con el tiempo, es decir, la dinámica de la *población* (similar a nuestro  $OE$ ) y la *organización* (similar al  $OS$  que proponemos) del sistema. Por otro lado, para modelar la estructura de una organización, la propuesta OperA [Dig03] también ofrece una descripción formal de una organización y presenta relaciones entre roles que son similares a nuestra propuesta. La relación de supervisión (*sup*) de nuestra propuesta es similar a la combinación de las relaciones de poder y autorización de OperA. La relación de supervisión expresa que un agente puede delegar sus objetivos a un agente subordinado, mientras que la relación de poder expresa que un agente es capaz de solicitar a un agente subordinado que alcance sus objetivos (la relación de autorización expresa la relación de poder como una situación temporal). Adicionalmente, en OperA el objetivo que un agente subordinado puede tomar de un agente superior se determina por el tipo de relación existente entre roles, y tiene tres posibilidades: (i) relación jerárquica, donde el agente superior delega un objetivo al agente subordinado; (ii) relación de mercado, donde el agente subordinado puja por un objetivo al agente superior; y (iii) relación de red, donde tanto el agente subordinado como el superior pueden solicitarse un objetivo entre ellos. La jerarquía entre roles de la organización se establece usando estas relaciones. Nuestra propuesta define esta jerarquía usando la relación *Hier*.

Hübner *et al* [HSB02] presentan otra aproximación para modelar la estructura de la organización. Su propuesta modela los tres niveles principales

de MOISE<sup>+</sup> (roles, relaciones de roles y grupos) por medio de tres niveles estructurales de la organización: (i) el *nivel individual*, formado por los roles de la organización y presenta relaciones de herencia entre roles (similar a nuestra relación *Hier*); (ii) el *nivel social*, que se construye de relaciones de enlace entre roles que se pueden clasificar como *acq* (conocimiento), si los agentes que juegan un rol origen pueden tener una representación de los agentes que juegan un rol destino, *com* (comunicación, similar a la relación *inf* que proponemos), si los agentes son capaces de comunicarse entre ellos, y *aut* (autoridad) si los agentes origen pueden tener autoridad sobre los agentes destino. Este último tipo de enlace es similar a una combinación de las relaciones *mon* and *sup* que proponemos; y (iii) el *nivel colectivo*, que define los *grupos* de los agentes de la organización, estableciendo la *compatibilidad* entre roles y su *cardinalidad*.

La propuesta que presentamos para modelar el entorno es similar a la que se presenta en el metamodelo de SODA [MODR06], que usa el concepto de workspace, la porción de entorno que un agente puede percibir, en vez de la entidad OU. En SODA, un agente sólo es capaz de usar los artefactos localizados en su workspace.

Finalmente, nuestra Entidad Organizativa puede compararse con otras propuestas. Para diferenciar entre una Unidad Organizativa y un Grupo, es posible comparar esta propuesta con la propuesta de MOISE<sup>+</sup>. Un grupo especificado se define en MOISE<sup>+</sup> como una 'especificación de grupo', mientras que nuestra propuesta lo define como una Unidad Organizativa. Por otro lado, la instanciación de un grupo se llama 'grupo' en ambas propuesta, la de MOISE<sup>+</sup> y la nuestra.

# Capítulo 6

## Conclusiones y trabajo futuro

En este último capítulo se recogen las principales conclusiones de este trabajo, las líneas futuras de investigación, así como las publicaciones relacionadas, producidas como resultado del trabajo de investigación realizado.

### 6.1. Aportaciones

De acuerdo a los objetivos presentados al inicio de la presente tesis, las principales aportaciones de la misma han sido:

- Se ha definido el concepto de Sistema Multiagente Centrado en la Organización, destacando las dimensiones en las que se divide, y se han presentado los lenguajes de modelado organizativo (OML) más relevantes, tanto de la aproximación basada en metamodelos, en el que se ha hecho especial hincapié en el Virtual Organization Model (VOM), desarrollado en nuestro grupo de investigación, como en los modelos formales más importantes del campo de los SMA.
- Se ha aportado una nueva taxonomía para clasificar los artefactos que pueblan el entorno de un SMA. Esta taxonomía, al contrario que las existentes, se basa en la funcionalidad que aportan los artefactos.
- Se ha revisado el concepto de mecanismo organizativo. Además, hemos modelado estos mecanismos como artefactos para facilitar su uso por parte de los diseñadores de SMA.
- Se ha propuesto una nueva versión del Virtual Organization Model, cuya mayor modificación ha sido en el entorno, para adoptar los conceptos

de artefacto, con especial atención a los artefactos de mecanismos organizativos, y workspace. Además, se han incluido diversas modificaciones que buscan mejorar el metamodelo.

- El concepto de adaptación en sistemas multiagente ha sido analizado, haciendo un estudio de la materia y presentando diferentes puntos de vista. Estos conceptos serán muy importantes en el futuro más próximo de nuestra investigación.

## 6.2. Trabajo futuro

Como trabajo futuro se pretende plantear un método de desarrollo de organizaciones virtuales adaptativas incorporando los artefactos y mecanismos adecuados para dar soporte al desarrollo de este tipo de sistemas. Para ello es necesario tener en cuenta que el rendimiento y funcionamiento adecuado de una organización puede verse afectado tanto por fuerzas externas como por internas. Las 'fuerzas externas' son debidas a los factores del entorno, es decir, a los cambios que se producen en las necesidades de los clientes/proveedores de la organización, a los cambios en los productos y/o recursos del sistema, etc. La organización debe adecuar su estructura para adaptarse a estos cambios en el entorno. Las 'fuerzas internas' son debidas a los comportamientos interesados de los miembros de la organización, que busquen más la satisfacción de sus propios intereses que los de la propia organización. En estos casos, el sistema debe aplicar medidas normativas o de corrección para reconducir el comportamiento de los miembros hacia los parámetros deseados. Es por ello que, durante el análisis y diseño del sistema, se debe tener en cuenta la posibilidad de adaptación y evolución de la organización en base a estos dos tipos de fuerzas, para que así el rendimiento global del sistema no se vea afectado negativamente por la existencia de las mismas.

Las recientes metodologías SMA orientadas a la organización se centran en los aspectos organizativos de la sociedad de agentes, así como en su regulación a través de normas, para proporcionar un enfoque de sistema abierto. En algunas metodologías SMA se emplean construcciones de tipo top-down para permitir comportamientos adaptativos en base a las fuerzas externas. Sin embargo, en la mayoría de las metodologías SMA no se contemplan las fuerzas internas/externas del sistema. Por tanto, se precisa la extensión de este tipo de metodologías, de modo que ofrezcan métodos y mecanismos al diseñador para facilitarle la identificación de las situaciones en las que los miembros de la organización deben modificar su comportamiento, sus relaciones estructurales, o incluso sus objetivos. La metodología deberá proporcionar guías o

métodos para diseñar los mecanismos de adaptación de la organización.

Este trabajo quedará enmarcado dentro del proyecto 'OVAMAH: Organizaciones Virtuales Adaptativas: Mecanismos, Arquitecturas y Herramientas' (referencia TIN2009-13839-C03-01), centrado en proveer de herramientas y metodologías que permitan a las Organizaciones Virtuales modificarse ante cambios en su entorno o sus requisitos. Las próximas actividades relacionadas con este proyecto son:

- Emplear la formalización propuesta en esta Tesis de Máster para describir las fuerzas internas y externas, además de los tipos de cambios y adaptación que deben ser tenidos en cuenta.
- Revisar el metamodelo VOM propuesto en esta Tesis de Máster para incluir aquellos aspectos de adaptación que hayan sido identificados.
- Modificar la metodología GORMAS [Arg08], asociada al metamodelo VOM, para reflejar todas estas modificaciones. La metodología deberá proporcionar guías para facilitar al diseñador la transformación o adaptación de la estructura de la organización en función de los tipos de fuerzas externas consideradas. Además, la metodología deberá facilitar al diseñador la identificación de los procesos en los que aparezcan las fuerzas internas, es decir, aquellos procesos donde los intereses particulares de los agentes puedan provocar comportamientos anómalos de la organización. Asimismo, deberá facilitar al diseñador los mecanismos y técnicas de tipo *self-organizing* para permitir establecer modificaciones sobre las relaciones de los miembros, así como redefinir las restricciones (normas) sobre sus comportamientos. Finalmente, la metodología deberá incorporar especificaciones de 'requisitos de adaptación' que faciliten al diseñador la identificación y diseño de las posibles transiciones estructurales de la organización.

### 6.3. Publicaciones relacionadas

En esta sección se presentan las publicaciones relacionadas con esta tesis de máster.

- S. Esparcia, R. Centeno, R. Hermoso and E. Argente. *Artifacting the Organizational Mechanisms: Adding Functionality in MAS Environments*. 2010 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-10) pp. In Press. (2010) (*CORE: C, CiteSeer<sup>x</sup>: 0.02 (375/581)*)

- S. Esparcia and E. Argente. *A functional taxonomy for artifacts*. 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS '10) pp. In Press. (2010) (*CORE: C*)
- S. Esparcia, V.Sánchez-Anguix, E. Argente, A. García-Fornes and V. Julian. *Integrating information extraction agents into a tourism recommender system*. 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010) pp. In Press. (2010) (*CORE: C*)
- V.Sánchez-Anguix, S. Esparcia, E. Argente, A. García-Fornes and V. Julian. *Collaborative information extraction for adaptive recommendations in a multiagent tourism recommender system*. 8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2010) Vol. 70 pp. 35-40. (2010) (*CORE: C*)
- S. Esparcia, E. Argente, V. Julian and V. Botti. *GORMAS: A methodological guideline for organizational-oriented open MAS*. Technical Report. Departamento de Sistemas Informaticos y Computacion. Universidad Politecnica de Valencia (2010)



# Bibliografía

- [ABC<sup>+</sup>08] R. Ali, V. Bryl, G. Cabri, M. Cossentino, F. Dalpiaz, P. Giorgini, A. Molesini, A. Omicini, M. Puviani, and V. Seidita. MEnSA Project - Methodologies for the Engineering of complex Software systems: Agent-based approach. Technical Report 1.2, UniTn, 2008.
- [ABD<sup>+</sup>07] R. Ali, V. Bryl, F. Dalpiaz, P. Giorgini, G. Cabri, A. Molesini, and A. Omicini. MEnSA Project - Methodologies for the Engineering of complex Software systems: Agent-based approach. Technical Report 1.1, UniTn, 2007.
- [ABFF<sup>+</sup>10] E. Argente, G. Beydoun, R. Fuentes-Fernandez, B. Henderson-Sellers, G. Low, and F. Migeon. Modelling with agents. In *Postproceedings of AOSE 2009*, page In Press. Springer, 2010.
- [AC09] E. Argente and R. Centeno. Design and analysis of organisational structures, 2009.
- [Ant04] R.J. Anthony. Emergence: A paradigm for robust and scalable distributed applications. In *Proceedings of IEEE International Conference on Autonomic Computing (ICAC'04)*, pages 132–139, 2004.
- [APDD08] H. Aldewereld, L. Penserini, F. Dignum, and V. Dignum. Regulating Organizations: The ALIVE Approach. *Proceedings of ReMoD 2008*, 2008.
- [Arg08] E. Argente. *GORMAS: Guías para el desarrollo de Sistemas multi-agente abiertos basados en organizaciones*. PhD thesis, Universidad Politecnica de Valencia, 2008.
- [ASM80] J.R. Abrial, S.A. Schuman, and B. Meyer. Specification language. *On the Construction of Programs: An advanced course*, pages 343–410, 1980.

- [BDL04] D. Basin, J. Doser, and T. Lodderstedt. Model driven security. *Engineering theories of software intensive systems*, pages 353–398, 2004.
- [BGG<sup>+</sup>04] P. Bresciani, P. Girogini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An agent-oriented software development methodology. *Autonomous Agent and Multi-Agent Systems*, 8:203–236, 2004.
- [BGPP03] C. Bernon, M.P. Gleizes, S. Peyruqueou, and G. Picard. Adelfe: A methodology for adaptive multi-agent systems engineering. *Engineering Societies in the Agents World III*, pages 70–81, 2003.
- [BLHS<sup>+</sup>09] G. Beydoun, G. Low, B. Henderson-Sellers, H. Mouratidis, J.J.G. Sanz, J. Pavón, and C. Gonzalez-Perez. FAML: a Generic Metamodel for MAS development. 2009.
- [BPR99] F. Bellifemine, A. Poggi, and G. Rimassa. JADE: A FIPA-compliant agent framework. In *Proc. of PAAM*, pages 97–108, 1999.
- [CAB10] N. Criado, E. Argente, and V. Botti. Rational Strategies for Autonomous Norm Adoption. In *Proc. COIN@AAMAS*, 2010.
- [CAJB09] N. Criado, E. Argente, V. Julian, and V. Botti. Designing virtual organizations. In *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS2009)*, volume 55 of *Advances in Soft Computing*, pages 440–449, 2009.
- [Cal09] J. Calta. A taxonomy of adaptive systems. In *7th European Workshop on Multi-Agent Systems (EUMAS 2009)*, pages 1–13, 2009.
- [CBHO09] R. Centeno, H. Billhardt, R. Hermoso, and S. Ossowski. Organising MAS: a formal model based on organisational mechanisms. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 740–746. ACM, 2009.
- [CGJ<sup>+</sup>09] C. Carrascosa, A. Giret, V. Julian, M. Rebollo, E. Argente, and V. Botti. Service oriented MAS: an open architecture. In *Proceedings of The 8th International Conference on Autonomous*

- Agents and Multiagent Systems-Volume 2*, pages 1291–1292. International Foundation for Autonomous Agents and Multiagent Systems, 2009.
- [Cla99] E.M. Clarke. Model checking. In *Foundations of Software Technology and Theoretical Computer Science*, page 54. Springer, 1999.
- [Cos05] M. Cossentino. From requirements to code with the passi methodology. *Agent Oriented Methodologies IV*, pages 79–106, 2005.
- [CSB05] L. Coutinho, J. Sichman, and O. Boissier. Modeling Organization in MAS: A Comparison of Models. In *First Workshop on Software Engineering for Agent-oriented Systems*, pages 1–10, 2005.
- [CZ06] L. Cernuzzi and F. Zambonelli. Dealing with adaptive multi-agent organizations in the gaia methodology. *Agent-Oriented Software Engineering VI*, pages 109–123, 2006.
- [CZ10] L. Cernuzzi and F. Zambonelli. Adaptive Organizational Changes in Agent-Oriented Methodologies. *The Knowledge Engineering Review*, 2010.
- [DD06a] V. Dignum and F. Dignum. A landscape of agent systems for the real world. Technical Report 44-CS-2006-061, Institute of Information and Computing Sciences, Utrecht University, 2006.
- [DD06b] V. Dignum and F. Dignum. Towards formal semantics for re-organization. 2006.
- [DD07] V. Dignum and F. Dignum. A logic for agent organizations. In *Proceedings of the 3rd Workshop on Formal Approaches to Multi-Agent Systems (FAMAS)*. Citeseer, 2007.
- [DeL04] S. DeLoach. The MASE methodology. *Methodologies and software engineering for agent systems*, pages 107–125, 2004.
- [DeL09] S. DeLoach. *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, chapter Organizational Model for Adaptive Complex Systems, pages 1–26. IGI Global, 2009.
- [Dig03] V. Dignum. *A model for organizational interaction: based on agents, founded in logic*. PhD thesis, Utrecht University, 2003.

- [Dig09] V. Dignum. *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*. Information Science Reference, 2009.
- [DMWD02] V. Dignum, J. Meyer, H. Wiegand, and F. Dignum. An organization-oriented model for agent societies. In *Proc. Int. Workshop on Regulated Agent-Based Social Systems (RASTA-02)*, 2002.
- [dRCD08] A. da Rocha Costa and G. Dimuro. Semantical concepts for a formal structural dynamics of situated multiagent systems. *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, pages 139–154, 2008.
- [dSBF10] V.T. da Silva, C. Braga, and K. Figueiredo. A Modeling Language to Model Norms. *Coordination, Organization, Institutions and Norms in Multi-Agent Systems@ AAMAS2010*, page 25, 2010.
- [DSD04] MV Dignum, E. Sonenberg, and FPM Dignum. Dynamic Reorganization of Agent Societies. In *Proceedings of Workshop on Coordination in Emergent Agent Societies*, 2004.
- [DW04] K.H. Dam and M. Winikoff. Comparing agent-oriented methodologies. In *Agent-Oriented Information Systems*, pages 78–93. Springer, 2004.
- [DWH05] T. De Wolf and T. Holvoet. Towards a methodology for engineering self-organising emergent systems. *Self-Organization and Autonomic Informatics (I)*, 135(1):18–34, 2005.
- [EA10] S. Esparcia and E. Argente. A functional taxonomy for artifacts. In *5th International Conference on Hybrid Artificial Intelligence Systems*, page In Press. Springer, 2010.
- [EAP97] A. Eardley, D. Avison, and P. Powell. Strategic information systems: An analysis of development techniques which seek to incorporate strategic flexibility. *Journal of Organizational Computing*, 7(1):57–77, 1997.
- [ECHA10] S. Esparcia, R. Centeno, R. Hermoso, and E. Argente. Artifacts the Organizational Mechanisms: Adding functionality

- in MAS Environments. In *2010 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, page In Press. IEEE, 2010.
- [Eme91] E.A. Emerson. Temporal and modal logic, Handbook of theoretical computer science (vol. B): formal models and semantics, 1991.
- [ERS<sup>+</sup>01] M. Esteva, J.A. Rodriguez, C. Sierra, P. Garcia, and J.L. Arcos. On the formal Specification of Electronic Institutions. *Agent Mediated Electronic Commerce*, LNAI 1991:126–147, 2001.
- [Est03] M. Esteva. Electronic Institutions: from specification to development. *IIIA PhD Monography*, 19, 2003.
- [FG98] J. Ferber and O. Gutknecht. Aalaadin: A meta-model for the analysis and design of organizations in multi-agent systems. In *Proc. 3rd Int. Conference of Multi-Agent Systems (ICMAS-98)*, pages 128–135, 1998.
- [FGM03] J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: an organizational view of multi-agent systems. In *Proc. AOSE*, pages 214–230, 2003.
- [GAG09] E. Garcia, E. Argente, and A. Giret. A modeling tool for service-oriented open multiagent systems modeling tool. In *The 12th International Conference on Principles of Practice in Multi-Agent Systems. PRIMA 2009*, volume 5925 of LNAI, pages 345–360. Springer-Verlag, 2009.
- [GBKD05] B. Gâteau, O. Boissier, D. Khadraoui, and E. Dubois. Moiseinst: An organizational model for specifying rights and duties of autonomous agents. In *Third European Workshop on Multi-Agent Systems (EUMAS 2005)*, pages 484–485. Citeseer, 2005.
- [GCG99] M.P. Gleizes, V. Camps, and P. Glize. A theory of emergent computation based on cooperative self-organization for adaptive artificial systems. In *Fourth European Congress of Systems Science*, 1999.
- [GDD<sup>+</sup>07] D. Grossi, F. Dignum, V. Dignum, M. Dastani, and L. Royakkers. Structural aspects of the evaluation of agent organizations. *Lecture Notes in Computer Science*, 4386:3, 2007.

- [GJR<sup>+</sup>09] A. Giret, V. Julian, M. Rebollo, E. Argente, C. Carrascosa, and V. Botti. An open architecture for service-oriented virtual organizations. In *Seventh international Workshop on Programming Multi-Agent Systems.PROMAS 2009*, pages 74–88, 2009.
- [GJW08] M. Ghijsen, W. Jansweijer, and B. Wielinga. Towards a framework for agent coordination and reorganization, agentcore. *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, pages 1–14, 2008.
- [GM97] N. Glaser and P. Morignot. The reorganization of societies of autonomous agents. *Multi-Agent Rationality*, pages 98–111, 1997.
- [GODOV08] J. Garcia-Ojeda, S. DeLoach, W. Oyenan, and J. Valenzuela. O-MaSE: a customizable approach to developing multiagent development processes. *Agent-Oriented Software Engineering VIII*, pages 1–15, 2008.
- [GPL07] J. Gonzalez-Palacios and M. Luck. Towards Compliance of Agents in Open Multi-agent Systems. *Software Engineering for Multi-Agent Systems V*, pages 132–147, 2007.
- [Gra59] P.P. Grassé. La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. la théorie de la stigmergie: Essai d’interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6(1):41–80, 1959.
- [Gru93] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.
- [GS98] H. Gazendam and J. Simons. An analysis of the concept of equilibrium in organization theory. In *Proceedings of the Computational and Mathematical Organization Theory Workshop*, 1998.
- [GVCO08] L. Gardelli, M. Viroli, M. Casadei, and A. Omicini. Designing self-organising environments with agents and artefacts: a simulation-driven approach. *International Journal of Agent-Oriented Software Engineering*, 2(2):171–195, 2008.

- [GVO07] L. Gardelli, M. Viroli, and A. Omicini. A.: Design patterns for self-organizing multiagent systems. *Proceedings of EEDAS*, 2007.
- [HBKR09] J.F. Hubner, O. Boissier, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. *Auton. Agents Multi-Agent Syst.*, 2009.
- [HBO10] R. Hermoso, H. Billhardt, and S. Ossowski. Role Evolution in Open Multi-Agent Systems as an Information Source for Trust. In *Proc. AAMAS*, 2010.
- [HBV08] J.F. Hubner, O. Boissier, and L. Vercoouter. Instrumenting multi-agent organisations with reputation artifacts. *Proc. COIN*, pages 96–110, 2008.
- [HMMF09] C. Hahn, C. Madrigal-Mora, and K. Fischer. A platform-independent metamodel for multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 18(2):239–266, 2009.
- [HRHL01] N. Howden, R. Ronnquist, A. Hodgson, and A. Lucas. JACK intelligent agents: Summary of an Agent Infrastructure. In *Proc. 5th Int. Conference on Autonomous Agents*, 2001.
- [HSB02] J.F. Hubner, J.S. Sichman, and O. Boissier. A model for the structural, functional, and deontic specification of organizations in multiagent systems. *Lecture notes in computer science*, pages 118–128, 2002.
- [HWHJ09] R. Haesevoets, D. Weyns, T. Holvoet, and W. Joosen. A formal model for self-adaptive and self-healing organizations. 2009.
- [JSTY07] C.M. Jonker, A. Sharpanskykh, J. Treur, and P.I. Yolum. A framework for formal modeling and analysis of organizations. *Applied Intelligence*, 27(1):49–66, 2007.
- [JT03] C.M. Jonker and J. Treur. A temporal-interactivist perspective on the dynamics of mental states. *Cognitive Systems Research*, 4(2):137–155, 2003.
- [KGJ09] R. Kota, N. Gibbins, and N.R. Jennings. Self-organising agent organisations. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 797–804. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

- [Lin00] J. Lind. Patterns in Agent-Oriented Software Engineering. 2000.
- [LM08] M. Luck and P. McBurney. Computing as interaction: agent and agreement technologies. In *Proc. DHMS*, pages 1–6, 2008.
- [Man96] M. Manzano. *Extensions of first order logic*. Cambridge Univ Pr, 1996.
- [MODR06] A. Molesini, A. Omicini, E. Denti, and A. Ricci. SODA: A roadmap to artefacts. *LNCS*, 3963:49, 2006.
- [OB07] J.S. Sichman O. Boissier, J.F. Hubner. Organization Oriented Programming, from closed to open organizations. In *Proc. Engineering Societies in the Agents World VI, Sixth International Workshop, ESAW06*, Lecture Notes in Computer Science, Dublin, Ireland, September 2007. Springer.
- [OMO08] E. Oliva, P. McBurney, and A. Omicini. Co-argumentation artifact for agent societies. *LNCS*, 4946:31, 2008.
- [ORV<sup>+</sup>04] A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, and L. Tumolini. Coordination artifacts: Environment-based coordination for intelligent agents. In *Proc. AAMAS*, pages 286–293, 2004.
- [ORV06] A. Omicini, A. Ricci, and M. Viroli. Agens Faber: Toward a theory of artefacts for MAS. *Electronic Notes in Theoretical Computer Science*, 150(3):21–36, 2006.
- [PG03] J. Pavon and J.J. Gomez. Agent Oriented Software Engineering with INGENIAS. In *Proc. CEEMAS 2003*, pages 394–403, 2003.
- [PHBG09] G. Picard, J.F. Hubner, O. Boissier, and M.P. Gleizes. Reorganisation and Self-organisation in Multi-Agent Systems. In *1st International Workshop on Organizational Modeling*, page 66, 2009.
- [PS06] V. Popova and A. Sharpanskykh. Process-Oriented Organization Modeling and Analysis Based on Constraints. Technical report, Citeseer, 2006.



- [PW07] H. V. D. Parunak and D. Weyns. Special issue on environments for multi-agent systems. *Auton. Agents Multi-Agent Syst.*, 14(1):1–4, February 2007.
- [RG97] A.S. Rao and M.P. Georgeff. Modeling rational agents within a BDI-architecture. *Readings in agents*, pages 317–328, 1997.
- [ROD06] R. Rubino, A. Omicini, and E. Denti. Computational institutions for modelling norm-regulated MAS: An approach based on coordination artifacts. *LNCS*, 3913:127, 2006.
- [ROV<sup>+</sup>07a] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva. Cognitive stigmergy: Towards a framework based on agents and artifacts. In *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 124–140. Springer, May 2007.
- [ROV<sup>+</sup>07b] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva. Cognitive stigmergy: Towards a framework based on agents and artifacts. *Environments for Multi-Agent Systems III*, pages 124–140, 2007.
- [RP09] A. Ricci and M. Piunti. Designing and Programming Agents' Environments in Multi-agent Systems. In *Proc. 11th EASSS*, pages 3–31, 2009.
- [RVO06] R. Ricci, M. Viroli, and A. Omicini. CArtAgO: An Infrastructure for Engineering Computational Environments. In *Proceedings E4MAS*, pages 102–119, 2006.
- [RVO07] A. Ricci, M. Viroli, and A. Omicini. Give agents their artifacts: the A&A approach for engineering working environments in MAS. In *Proc. AAMAS*, page 150, 2007.
- [Smo73] C. Smorynski. Applications of Kripke models. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, pages 324–391, 1973.
- [SV85] J.R. Searle and D. Vanderveken. *Foundations of illocutionary logic*. Cambridge Univ Pr, 1985.
- [VDP06] H. Van Dyke Parunak. A survey of environments and mechanisms for human-human stigmergy. *Environments for Multi-Agent Systems II*, pages 163–186, 2006.

- [vG91] J. P. van Gigch. *System Design Modeling and Metamodeling*. Plenum Press, 1991.
- [ZJW01] F. Zambonelli, N. Jennings, and M. Wooldridge. Organisational abstractions for the analysis and design of multi-agent systems. In *Agent-Oriented Software Engineering*, pages 407–422. Springer, 2001.
- [ZJW03] F. Zambonelli, N.R. Jennings, and M. Wooldridge. Developing Multiagent Systems: The Gaia Methodology. *ACM Transactions on Software Engineering and Methodology*, 12:317–370, 2003.