# EVALUACIÓN DE MÉTODOS Y TÉCNICAS PARA EL DESARROLLO DE SISTEMAS MULTIAGENTE.

Autor: María Emilia García Marqués
Directores: Dr. Adriana Giret Boggino
Dr. Vicente J. Botti Navarro

Para la obtención del Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital
Valencia, España
SEPTIEMBRE 2008

Fecha: **Septiembre 2008**


Autor: **María Emilia García Marqués**

Directores: **Dr. Adriana Giret Boggino**

**Dr. Vicente J. Botti Navarro**


Título: **Evaluación de métodos y técnicas para el desarrollo de sistemas multiagente.**


Departamento: **Sistemas Informáticos y Computación**

Universidad: **Universidad Politécnica de Valencia**


Mes: **Septiembre** Año: **2008**

Firma del Autor

# Índice general

# Índice de figuras

# Capítulo 1

# Introducción

Los sistemas basados en agentes son una de las áreas de investigación y desarrollo con mayor importancia de las que han emergido dentro de las tecnologías de información durante la década de los 90, y ofrecen soporte a muchos aspectos de las aplicaciones e infraestructuras de computación actuales. El crecimiento del interés en este campo es palpable ya que numerosos grupos de investigación están realizando un esfuerzo sobre estas áreas, llegando a obtener avances significativos tanto desde el punto de vista teórico como práctico [Wooldridge and Ciancarini, 2001; Bordini, Dastani, and Winikoff, 2007].

Los sistemas multiagente son sistemas complejos y distribuidos compuestos por entidades autónomas [Wooldridge, 1997]. Por ello, el desarrollo de sistemas multiagente es una tarea compleja que necesita el uso de técnicas de ingeniería del software para conseguir sistemas completos, robustos y funcionales en un tiempo razonable.

La ingeniería del software en los sistemas multiagente está basada y tiene muchas similitudes con la ingeniería del software tradicional, aunque tiene algunas diferencias debido a que ha de tener en cuenta todas las características específicas de los sistemas multiagente.

Debido a esto durante los últimos años han surgido numerosos métodos y herramientas para desarrollar sistemas multiagente, tales como: metodologías (Gaia [Wooldridge, Jennings, and Kinny, 2000], Tropos [Bresciani et

al., 2004], RT-Message [Julian and Botti, 2004], Anemona [Giret, Botti, and Valero, 2005], Prometheus [Padgham and Winikoff, 2004b], Ingenias [Pavon and Gomez, 2003]), entornos de desarrollo ([Arcos et al., 2005]), lenguajes de modelado (AUML [UML, 2008], AML [Trencansky and Cervenka, 2005]), herramientas de depuración ([Val et al., 2007]), plataformas de agentes (Jade [Bellifemine, Caire, and Greenwood, 2007]), lenguajes de programación (JACK [Jack agent platform, 2008]), etc.

Cada propuesta está orientada a diferentes aspectos, ofrece diferente funcionalidad, e incluso, en algunos casos, se basan en conceptos de agente o de sistema multiagente que difieren del resto en ciertos aspectos. Cada una de ellas cubre una o varias etapas del proceso de desarrollo, dándose el caso de que algunas de ellas sólo presentan una descripción teórica o guías de desarrollo sin ofrecer ninguna herramienta que permita aplicar esas guías teóricas.

Esta situación hace que la selección entre una u otra herramienta de desarrollo de sistemas multiagente sea una tarea complicada y en muchos casos el desarrollador seleccione la herramienta que tiene más a mano sin tener en cuenta cuál sería más adecuada para programar una aplicación en concreto.

Para intentar subsanar este problema, se han realizado numerosos estudios sobre la evaluación de métodos y herramientas de sistemas multiagente [Braubach, Pokahr, and Lamersdorf, 2008; Eiter and Mascardi, 2002; Sudeikat et al., 2005]. A pesar de esto y como se analizará en la Sección 2, actualmente no existe ninguna herramienta de evaluación que permita la evaluación de todos los métodos y herramientas que son necesarias en el desarrollo de sistemas multiagente.

En este trabajo se abordará este problema contribuyendo al estado del arte con un estudio detallado de los criterios de evaluación de métodos, técnicas y herramientas para desarrollar sistemas multiagente. De este estudio se derivan: (1) un cuestionario completo y no ambiguo que integra un alista de criterios que ayudan en la selección y evaluación de herramientas de desarrollo de sistemas multiagente; y (2) una herramienta de evaluación

de métodos y herramientas para desarrollar sistemas multiagente, llamada Masev que simplifica y facilita el proceso de evaluación. Masev analiza todas las etapas del proceso de desarrollo, teniendo en cuenta las facilidades que se ofrecen para pasar de los requisitos del problema al modelado y de los modelos conceptuales al código final. Masev también evalúa el desarrollo de sistemas multiagente organizativos y la integración entre sistemas multiagente y servicios. Estos son dos tendencias de los sistemas multiagente que están tomando mucha importancia en los últimos años [Criado et al., 2007; Dignum et al., 2007; Bade et al., 2008; Tinnemeier, Dastani, and Meyer, 2008; Ossowski et al., 2007; Greenwood et al., 2007].

## 1.1. Origen del proyecto

Este proyecto se ha realizado dentro del Grupo de Investigación de Tecnología Informática-Inteligencia Artificial del Departamento DSIC (GTIA).

El trabajo presentado en este trabajo se enmarca dentro del proyecto THOMAS y del proyecto Consolider Agreement Technologies 2007. El proyecto THOMAS (MeTHods, Techniques and Tools for Open Multi-Agent Systems) consiste en la investigación y desarrollo de la tecnología basada en agentes/sistemas multiagente necesaria para el desarrollo de organizaciones virtuales en entornos abiertos (sistemas abiertos de servicios basados en agentes). El proyecto Agreement Technologies tiene por objetivo desarrollar modelos, entornos, métodos y algoritmos para construir sistemas distribuidos, dinámicos y escalables que consideren conceptos y técnicas de seguridad, reputación y argumentación.

## 1.2. Objetivos

El principal objetivo de este trabajo es realizar un estudio detallado de la evaluación en el campo de los sistemas multiagente y específicamente contribuir en el análisis, evaluación y comparación de métodos y herramientas para desarrollar sistemas multiagente.

Este objetivo global se deriva en los siguientes subobjetivos:

- **Análisis:**
  - Analizar el estado del arte de la ingeniería del software en los sistemas multiagente.
  - Analizar las técnicas y herramientas existentes para el desarrollo de sistemas multiagente. Este punto incluirá la utilización de algunas de estas herramientas para el modelado, la implementación y la evaluación de una aplicación basada en agentes. Este desarrollo permitirá un mayor entendimiento de los problemas y necesidades que surgen al desarrollar este tipo de sistemas.
  - Analizar el estado del arte en la evaluación de métodos y entornos de desarrollo de sistemas multiagente.
  - Analizar las características propias de los sistemas multiagente organizacionales.
  - Analizar las características propias de los sistemas multiagente orientados a servicios, es decir, de aquellos sistemas que integran la tecnología multiagente y de servicios.
- **Definición de criterios:**
  - Determinar qué características y funcionalidades son más importantes para desarrollar sistemas multiagente, es decir, determinar los criterios de evaluación para cada uno de los métodos y herramientas implicadas en el desarrollo de este tipo de sistemas.

    Los criterios de evaluación deberán cubrir todas las etapas del proceso de desarrollo, desde la extracción de requisitos hasta la implementación, ejecución y monitorización final. Por ello se deberán analizar las características que definen las metodologías, lenguajes de modelado, herramientas de modelado, entornos de implementación.

- Definir las necesidades y requisitos que se derivan de la introducción del concepto de sistemas organizacionales en los sistemas multiagente.
- Definir las necesidades y requisitos que se derivan de la integración de los sistemas multiagente y de los servicios.

▪ **Implementación de la herramienta de evaluación:**

- Definir un cuestionario que permita la evaluación de los métodos y herramientas para desarrollar sistemas multiagente, basado en los criterios seleccionados anteriormente.
- Desarrollar una aplicación online que permita:
  - Evaluar metodologías, lenguajes de modelado, herramientas de modelado, herramientas de implementación en base al cuestionario especificado.
  - Almacenar y gestionar la información de las evaluaciones.
  - Comparar diferentes técnicas y métodos en función de la información obtenida en las evaluaciones.

## 1.3. Metodología del proceso de evaluación

La evaluación de un método o de una herramienta software tiene por objetivo determinar en qué medida dicho producto cumple determinadas características [Punter et al., 2004].

La organización ISO [iso, 2008] trabaja para el desarrollo de estándares tales como [ISO, 2001; 2001] que definen las características y sus subcaracterísticas más importantes para determinar la calidad de un producto desde el punto de vista de la ingeniería del software.

A su vez, ISO define los estándares [ISO, 1999; 2005] que estandarizan el proceso de evaluación en sí mismo. El objetivo de dicha estandarización es conseguir procesos de evaluación lo más objetivos posible y reproducibles.

ISO en [ISO, 1999] establece 4 fases que todo proceso de evaluación debe seguir:

- **Establecer los requisitos de la evaluación:** En esta fase se define el propósito de la evaluación y los tipos de productos a evaluar. Por último se define el modelo de calidad, es decir, la lista de características a evaluar.

- **Especificar la evaluación:** En esta fase se asocia a cada una de las características definidas en la fase anterior una escala cuantitativa (una puntuación a cada una de las respuestas posibles) y un método que permita obtener resultados numéricos de la evaluación.

- **Diseñar la evaluación:** En esta fase se diseña el plan de evaluación, es decir, se define qué recursos son necesarios, se documentan los procesos que va a seguir el evaluador, etc.

- **Ejecutar la evaluación:** Finalmente se ejecuta la evaluación siguiendo el plan definido en la fase anterior y se analizan los resultados obtenidos.

En [Punter et al., 2004] se hace una revisión crítica de este estándar. Este trabajo plantea una nueva división en fases descomponiendo cada una de las fases anteriores en subfases y otorgando mayor importancia al establecimiento y priorización de los objetivos y los grupos de interés a los que se dirige la evaluación. De igual modo, estudia y propone diferentes puntos de retroalimentación entre las fases.

Basándonos en los estándares comentados, definimos el proceso de desarrollo del entorno de evaluación presentado en este trabajo. Como se puede observar en la Figura 1.1, el proceso comienza analizando el estado del arte y realizando diferentes estudios (capítulo 2) de los cuales se derivan la definición de los criterios de evaluación (capítulo 3) y la métrica a utilizar (capítulo 3.4). Posteriormente se diseña la evaluación a través del diseño y la implementación de la herramienta online Masev (capítulo 4). Finalmente

se establece el proceso de ejecución de la evaluación, del que se muestra un ejemplo en la sección 4.4.

La fase de actualización es accesible desde cualquier punto del proceso. Esto se debe a que conforme se va pasando por las diferentes fases, e incluso cuando se obtienen los resultados de las evaluaciones, el conocimiento sobre necesidades y características especiales en cada uno de los ámbitos va aumentando. Otra razón es que se pretende que el entorno de evaluación perdure en el tiempo y que sea dinámico adaptándose a los nuevos requerimientos y técnicas.

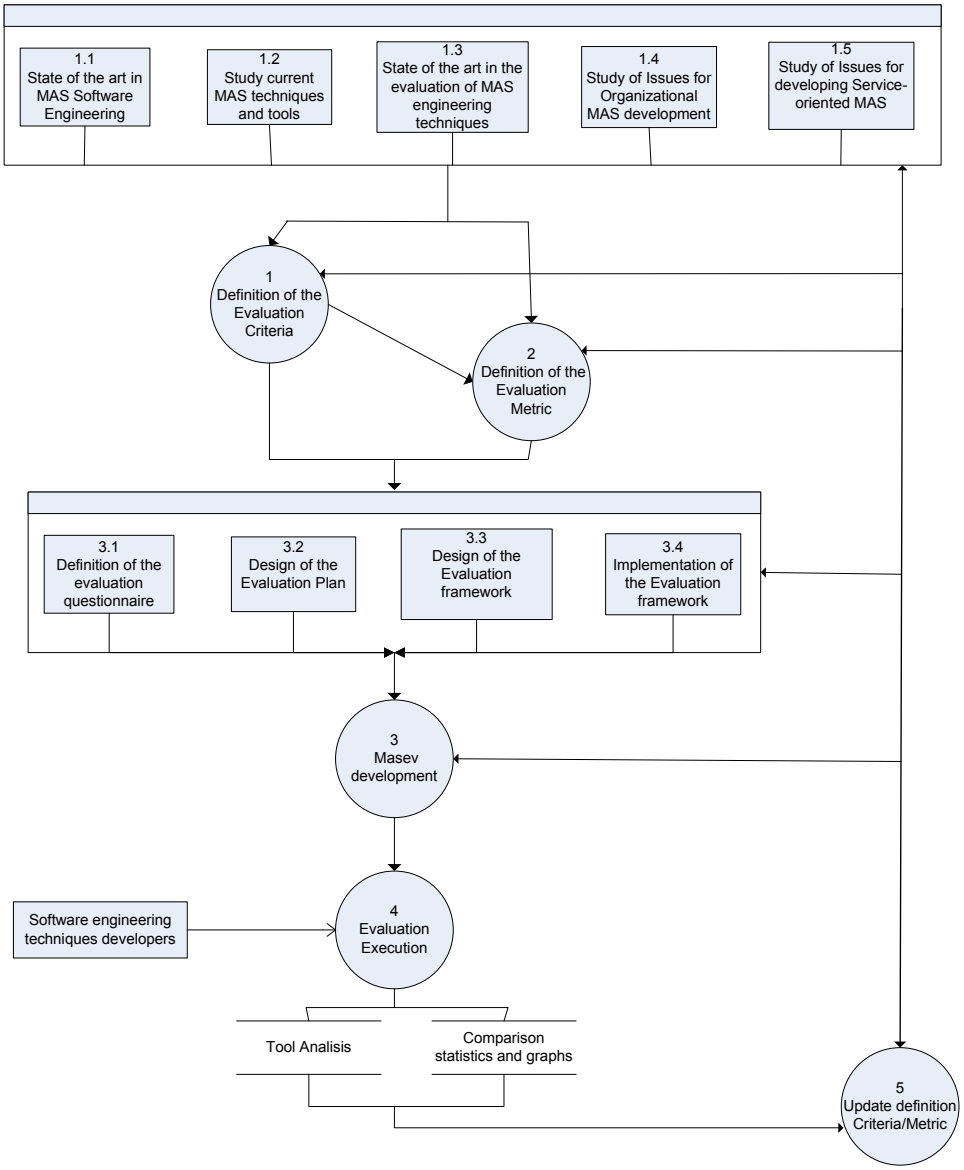


Figura 1.1: Proceso de desarrollo de Masev

# Capítulo 2

# Antecedentes

En este capítulo se van a analizar los antecedentes y estudios previos a este trabajo. Como base para este proyecto se han analizado los siguientes aspectos (Figura 1.1): (1) el estado del arte de la ingeniería del software en los sistemas multiagente; (2) estudio de las técnicas y herramientas de desarrollo de sistemas multiagente; (3) estado del arte de la evaluación de las técnicas de la ingeniería del software en sistemas multiagente; (4) estudio de las características necesarias para desarrollar sistemas multiagente organizacionales; (5) estudio de las características necesarias para sistemas que integren agentes y servicios.

## 2.1. Ingeniería del software en sistemas multiagente

Con el objetivo de fundamentar este trabajo se ha estudiado el estado del arte en la ingeniería del software en los sistemas multiagente.

Existen numerosos trabajos que abordan esta temática [Padgham and Winikoff, 2004a; Bordini, Dastani, and Winikoff, 2007; Wooldridge and Ciancarini, 2001; P et al., 2000; Bernon, Cossentino, and Pavón, 2005; Dastani et al., 2004; Xue, Zeng, and Liding, 2006; Braubach, Pokahr, and

Lamersdorf, 2006; Dignum et al., 2007]. De este análisis se extrajeron muchos de los criterios de evaluación presentados en el Capítulo 3. También se observó que era importante analizar todas las etapas del proceso de desarrollo y las herramientas implicadas en dichas etapas así cómo la relación entre ellas. Esta importancia se debe a que uno de los mayores problemas de los sistemas multiagente actuales es la gran diferencia entre lo que se define teóricamente y lo que finalmente se puede implementar.

A su vez, se han estudiado diferentes metodologías [Julian, 2002; Nwana et al., 1999; Giorgini et al., 2005; Pavon, Gomez-Sanz, and Fuentes, 2005; Hubner, Sichman, and Boissier, 2006], lenguajes de modelado [Trencansky and Cervenka, 2005], herramientas de modelado [Grasia, 2005; sta, 2008], herramientas de implementación [ecl, 2008], lenguajes de programación [Jack agent platform, 2008; Howden et al., 2001], entornos completos de desarrollo [Arcos et al., 2005; Sierra et al., 2007; Hao, W., and Zhang, 2005] y plataformas de agentes[Escriva et al., 2006; Bellifemine, Caire, and Greenwood, 2007].

Con el propósito de profundizar en este campo se diseñaron, modelaron e implementaron diferentes aplicaciones basadas en sistemas multiagente. La de mayor relevancia es una aplicación para la planificación y monitorización de la producción de una empresa cerámica [Garcia et al., 2008; Valero et al., 2007; 2006]. El análisis y el diseño se realizó utilizando la metodología Ingenias [Pavon, Gomez-Sanz, and Fuentes, 2005]. La implementación se realizó utilizando el entorno Eclipse [ecl, 2008] y la plataforma de ejecución destino escogida fue Jade [Bellifemine, Caire, and Greenwood, 2007].

En el momento de seleccionar qué métodos y herramientas usar para el desarrollo de esta aplicaciones se observó que a pesar de todos los estudios sobre evaluación de sistemas de desarrollo de agentes, no había ninguna herramienta que permitiera de forma sencilla comparar las opciones existentes, permitiendo así la selección entre uno u otro en base a unos criterios homogéneos. Al final la selección vino determinada por la experiencia previa de otros miembros del grupo de investigación GTI-IA.

A su vez, con el objetivo de tener una visión más amplia del desarrollo de

este tipo de sistemas se implementó sobre la plataforma Magentix [Alberola et al., 2007; Such et al., 2007] un sistema multiagente capaz de recomendar servicios turísticos a través de una aplicación online. Esta aplicación había sido implementada anteriormente utilizando la plataforma Jade [Lopez et al., 2007b; 2007a; Lopez, Bustos, and Julian, 2007], por lo que se pudieron observar las diferencias a nivel de programación y de funcionalidad entre ambas plataformas.

Por último, se estudiaron los diferentes sistemas de monitorización de agentes y se implementó una herramienta de depuración postmortem cuyos detalles se pueden consultar en [Val et al., 2007].

El modelado y la implementación de estas aplicaciones permitió analizar desde el punto de vista del diseñador y del programador el proceso de desarrollo de este tipo de sistemas. Esto ha facilitado la detección de muchos criterios de evaluación que a nivel teórico no parecen relevantes pero que a nivel práctico resultan de gran utilidad.

## 2.2. Evaluación de la ingeniería del software

Esta sección tiene como objetivo describir y detectar carencias en los estudios previos sobre la evaluación y comparación de métodos y técnicas para desarrollar sistemas multiagente.

La técnica más popular es la basada en la detección, estudio y comparación de ciertas características:

Julian y Botti [Julian and Botti, 2003] comparan algunas metodologías y expanden una de ellas para permitir el modelado de entornos de tiempo real.

Cernuzzi y Rossi [Cernuzzi and Rossi, 2002] presentan una lista de criterios para evaluar el modelado del análisis y el diseño de sistmas multiagente que amplían los presentados en [Shehory and Sturm, 2001]. A su vez, presentan una serie de fórmulas para obtener resultados numéricos de la evaluación.

Shehory y Sturm [Sturm and Shehory, 2004] realizan una evaluación

basada en características de las metodologías de agentes. Este trabajo revisa y amplía el presentado en [Shehory and Sturm, 2001]. Su criterio de evaluación incluye características derivadas de la ingeniería del software tradicional y características directamente relacionadas con el concepto de agente y su funcionamiento.

Lin y Kavi [Lin et al., 2007] proponen un framework para evaluar metodologías de agentes a dos niveles distintos. El nivel más superficial se encarga de determinar si una determinada característica está soportada por la metodología. En el nivel de detalle propone cuestiones concernientes a la relación entre el requisito y la funcionalidad ofrecida por la metodología aportando puntos para la comparación.

Los trabajos comentados centran sus esfuerzos en el análisis de metodologías, pero no tienen en cuenta las herramientas de modelado necesarias para llevar a cabo ese diseño, ni el resto de herramientas que son necesarias para implementar sistemas multiagente. El análisis de dichas herramientas es una parte muy importante de la evaluación, debido a que por muy bien definida que esté una metodología, esta pierde gran parte de su funcionalidad si no hay ninguna herramienta que la soporte.

Además, estos trabajos no analizan aspectos económicos (como la disponibilidad de documentación, ejemplos, etc de las metodologías estudiadas) y tampoco analizan las características necesarias para diseñar sistemas multiagente organizacionales, agentes móbiles, etc.

Eiter y Mascardi [Eiter and Mascardi, 2002] analizan entornos completos para desarrollar agentes software. Ellos proporcionan una metodología y unas guías generales para seleccionar un entorno de desarrollo u otro. Su lista de criterios incluye características propias de los agentes derivadas del concepto de agente, cómo soportan los requisitos propios de la ingeniería del software tradicional, las características necesarias para implementar agentes y sistemas complejos, características técnicas de los entornos de desarrollo y por último aspectos económicos de estos.

Bitting y Carter [Bitting, Carter, and Ghorbani, 2003] usan el criterio

establecido en [Eiter and Mascardi, 2002] para analizar 5 entornos de desarrollo. Bitting y Carter añaden una evaluación cuantitativa con el propósito de obtener resultados más objetivos. Realmente, en este caso la objetividad de los resultados obtenidos viene dada por la objetividad del evaluador al clasificar los elementos estudiados. Por ello, la evaluación cuantitativa no consigue resultados más objetivos, pero sí resultados más rápidos de entender y comparar.

Sudeikat y Braunch [Sudeikat et al., 2005] abordan y evalúan la problemática de la distancia entre el modelado y la implementación final. Como se demuestra en este artículo, y en artículos posteriores de la misma temática [Xue, Zeng, and Liding, 2006; Bordini, Dastani, and Winikoff, 2007], este es uno de los grandes problemas de la tecnología de agentes en la actualidad, ya que se diseñan modelos, métodos y arquitecturas muy abstractas y complejas que después no tienen correspondencia directa en la implementación.

Braubach y Pokahr [Braubach, Pokahr, and Lamersdorf, 2008] presentan un catálogo universal para evaluar herramientas de desarrollo de agentes heterogéneas. Este trabajo presenta una evaluación interesante respecto a las características propias de la ingeniería del software basada en ISO 9126-1 [ISO, 2001] y en ISO 9241 [ISO/IEC, 2006]. Pero al ser una evaluación genérica para todo tipo de herramientas no tiene en cuenta características propias de las metodologías ni de los entornos de desarrollo de agentes.

Trabajos como [Dam, 2003; Tran and Low, 2005], no sólo proporcionan una lista con los conceptos y características a analizar, sino que estructuran estos criterios en forma de cuestionario. El uso de cuestionarios fuerza a que los análisis sean más concretos y fáciles de comparar. De igual modo, reducen el tiempo de evaluación y simplifican esta tarea.

La principal carencia de [Dam, 2003; Tran and Low, 2005] es que se limitan a comparar metodologías y no tienen en cuenta otras herramientas y técnicas necesarias para el desarrollo de sistemas multiagente. Además, tampoco tienen en cuenta la relación entre el modelado y la implementación final.

## 2.3. Sistemas multiagente organizacionales

Los sistemas multiagente organizacionales (OMAS) están emergiendo rápidamente como un paradigma para desarrollar sistemas complejos. Dicho paradigma ha sido usado satisfactoriamente para desarrollar sistemas de agentes [Boissier et al., 2006; Ferber, Gutknecht, and Michel, 2004]. Una de las ventajas es que este tipo de sistemas permiten el modelado con un gran nivel de abstracción, por lo que la distancia entre el mundo real y los modelos se reduce considerablemente. De igual modo, este paradigma ofrece facilidades para implementar entornos abiertos, dinámicos y heterogéneos [Mao and Yu, 2005].

Este tipo de sistemas implica nuevos requisitos en los sistemas multiagente tradicionales, tanto en la forma de modelar como en la tecnología a usar ya que se debe incluir la integración de la perspectiva organizacional y la individual y la dinamicidad y la adaptabilidad a cambios en el entorno [Dignum and Dignum, 2006].

Las técnicas de la ingeniería del software para los sistemas multiagente tradicionales no son suficientes para desarrollar este tipo de sistemas. Nuevos métodos, herramientas y entornos de desarrollo son necesarios. En los últimos años se han desarrollado diferentes metodologías [Argente, Julian, and Botti, 2006; Ferber, Gutknecht, and Michel, 2004; Giorgini, Kolp, and Mylopoulos, 2006; Hubner, Sichman, and Boissier, 2006], técnicas de modelado [Horling and Lesser, 2005; Arcos et al., 2005] y plataformas [Argente et al., 2007; Esteva et al., 2004] orientadas a la organización.

Sin embargo, cada propuesta usa su propia terminología y ofrece distinta funcionalidad por lo que es muy difícil para los desarrolladores seleccionar entre una u otra. Muchos desarrolladores tienen dudas incluso en cómo traducir los conceptos organizacionales en entidades ejecutables de la aplicación [Boissier, Hübner, and Sichman, 2007] y qué plataforma y requisitos de implementación son necesarios para desarrollar sus modelos.

### 2.3.1. Ingeniería del software en sistemas multiagente organizacionales

Las organizaciones pueden ser materializadas en los sistemas multiagente de diferentes maneras [Boissier, Hübner, and Sichman, 2007]: (1) los desarrolladores no definen explícitamente la estructura de la organización, pero los agentes del sistema se comportan siguiendo un patrón organizativo [Pavon, Gomez-Sanz, and Fuentes, 2005]; (2) se especifica la estructura de la organización, pero los agentes no son conscientes de estar dentro de una organización y no razonan sobre ella; (3) cada agente es consciente de que pertenece a una organización y tiene una representación interna de los patrones de cooperación que sigue voluntariamente cuando quiere; (4) los agentes tienen una representación explícita de la organización. Algunos ejemplos de la clasificación 4 son S-Moise+ [Hubner, Sichman, and Boissier, 2006] y Ameli (Instituciones electrónicas) [Esteva et al., 2004]. Estos representan explícitamente la organización y tienen arquitecturas similares. Siguen una arquitectura basada en 3 niveles: el nivel de aplicación que está formado por los agentes autónomos de la aplicación, el nivel social que asegura que las interacciones sigan las normas de la organización, el nivel de comunicación que permite la comunicación entre agentes. Los agentes de la aplicación son los responsables de alcanzar tanto sus objetivos individuales como los objetivos de la organización.

En la actualidad la mayor parte de los trabajos relacionados con los sitemas multiagente organizacionales abordan las fases de análisis y diseño, pero existen pocas aproximaciones que aborden la implementación y la ejecución de este tipo de sistemas. A pesar de eso, existen plataformas orientadas a la organización tales como Spade [Argente et al., 2007] o Jack Teams [Software, 2004] y entornos completos de desarrollo como las Instituciones Electrónicas [Arcos et al., 2005]. Estos sistemas están en constante evolución y constantemente surgen nuevas aproximaciones y tecnologías específicas.

## 2.4. Sistemas multiagente orientados a servicios

Las arquitecturas orientadas a servicios se están extendiendo día a día, consisten en un conjunto de servicios independientes que se comunican usando estándares y protocolos de intercambio de mensajes. Prueba de la creciente importancia de este tipo de sistemas es que muchas empresas importantes como IBM, Microsoft, Intel, HP o Sun Microsystems están desarrollando herramientas para implementar estos sistemas y aplicaciones basadas en ellas.

Los objetivos de las arquitecturas orientadas a servicios y los de los sistemas multiagente son similares, es decir, ambos intentan crear sistemas distribuidos y flexibles compuestos por entidades independientes que interaccionan unos con los otros. A pesar de estas similitudes, la tecnología que usan es diferente. Los estándares y los protocolos de intercambio de mensajes son distintos por lo que ambas arquitecturas no pueden interaccionar directamente.

Numerosos estudios, tales como [Huhns et al., 2005; OMG, 2002; P.Singh and N.Huhns, 2005], han estudiado la interacción entre estas arquitecturas y han demostrado que de su interacción se obtienen muchos beneficios. Los servicios ofrecen una infraestructura e interoperabilidad bien definida, mientras que los agentes tratan de proporcionar inteligencia y capacidades sociales (confianza, reputación, etc) a las aplicaciones. Por esto, la integración entre agentes y servicios mejora la flexibilidad, interoperabilidad y funcionalidad de las aplicaciones.

Actualmente se están desarrollando metodologías, entornos de desarrollo y todo tipo de herramientas de ingeniería del software para desarrollar sistemas multiagente orientados a servicios. Cada aproximación tiene su propio mecanismo de integración, lenguaje de comunicación, incluso diferentes conceptos de agentes y servicios.

### 2.4.1. Ingeniería del software en sistemas orientados a servicios

Se basa en la ingeniería del software tradicional, pero tiene en cuenta las características específicas de los sistemas orientados a servicios.

Estos sistemas deben ser colaborativos ya que las aplicaciones orientadas a servicios suelen serlo. Los servicios clientes, proveedores y brokers colaboran para invocar, buscar, registrar y proporcionar servicios. Los sistemas pueden componerse en tiempo de ejecución por lo que gran parte de la ingeniería de este tipo de sistemas debe hacerse online. Los servicios están orientados para ser reusados por lo que es importante que su modelado sea independiente de la plataforma.

Actualmente se está estudiando en la ingeniería del software de este tipo de sistemas [Tsai et al., 2007; Papazoglou et al., 2006; Tsai, 2005]. Estos trabajos están concentrados en desarrollar metodologías para sistemas orientados a servicios y su modelado. En la primera línea de investigación los trabajos se concentran en cómo proporcionar suficientes principios y guías para especificar, construir, refinar y actualizar coreografías de procesos de negocio a partir de un conjunto de servicios web [Witwicki and Durfee, 2008; Papazoglou and van den Heuvel, 2006]. En la segunda línea de investigación los trabajos se concentran en diseñar modelos orientados a objetivos [Rolland, Souveyet, and Kraeim, 2008; Penserini et al., 2006]. En la segunda línea de investigación algunos trabajos se concentran en desarrollar este tipo de sistemas basándose en sistemas orientados al objetivo [Rolland, Souveyet, and Kraeim, 2008; Penserini et al., 2006].

## 2.5. Conclusiones

En este capítulo se ha hecho una breve introducción al estado del arte en la evaluación de las técnicas y métodos para desarrollar sistemas multiagente. De dicho estudio se puede concluir que es necesario el desarrollo de un sistema de evaluación completo que evalúe todas las etapas del proceso

de desarrollo.

De igual modo se han introducido los sistemas multiagente organizacionales y los sistemas multiagente orientados a servicios. Ambos paradigmas están en constante auge y aportan muchos beneficios en el desarrollo de sistemas multiagente. Estos sistemas comparten muchas características con los sistemas multiagent tradicionales, pero tienen requisitos específicos. Por este motivo, la evaluación de las herramientas de desarrollo de este tipo de sistemas debe incluir la evaluación de dichas características específicas.

# Capítulo 3

# Criterios de evaluación

En este capítulo se describe el entorno de evaluación, análisis y comparación presentado en este trabajo.

Este trabajo se basa en el estudio y comparación de los elementos y técnicas más relevantes a la hora de diseñar e implementar sistemas multiagente en función de ciertos criterios de evaluación.

Dicho conjunto de criterios tiene en cuenta no sólo características propias de la ingeniería del software clásica, sino que también considera las propiedades específicas del desarrollo de sistemas multiagente.

La selección de estos criterios viene derivada del estudio presentado en el capítulo 2.

El objetivo de estos criterios es evaluar todas las etapas del proceso de desarrollo, por ello se han tenido en cuenta los métodos y herramientas necesarias para diseñar e implementar sistemas multiagente desde la etapa de extracción de requisitos hasta su posterior implementación.

Debido a esto, los criterios se han agrupado principalmente entorno a qué métodos y herramientas participan en el desarrollo de estos sistemas (ver Figura 3.1).

El objetivo principal de este trabajo es ofrecer un entorno de evaluación completo y fácil de usar. Por ello los criterios de evaluación se presentan en forma de cuestionario.

El uso de cuestionarios agiliza el proceso de evaluación disminuyendo el

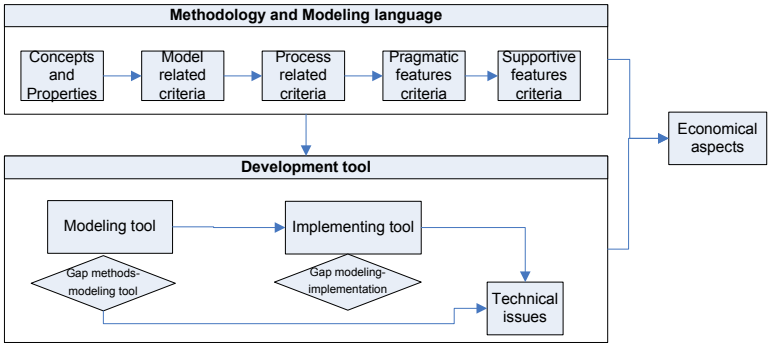


Figura 3.1: Clasificación de los criterios

tiempo que los usuarios necesitan para analizar sus métodos o herramientas. A su vez ayuda a concretizar y homogeneizar las respuestas, facilitando así la comparación entre herramientas.

Con la intención de darle la mayor difusión posible y de que el mayor número de métodos y herramientas sean evaluados, el cuestionario se presenta en inglés.

El resto del capítulo se estructura de la siguiente forma: la sección 3.1 describe el método de evaluación presentado para evaluar y comparar métodos y herramientas de sistemas multiagente tradicionales; la sección 3.2 describe los criterios para analizar cómo los métodos y herramientas soportan el desarrollo de sistemas multiagente organizacionales; la sección 3.3 describe los criterios para analizar cómo los métodos y herramientas soportan el desarrollo de sistemas multiagente orientados a servicios.

Debido al gran número de criterios de evaluación presentados en este trabajo, estos no se van a comentar uno a uno sino que para cada grupo de criterios derivado de la clasificación mostrada al principio de cada sección, se hará una breve introducción explicando qué se va a analizar y resaltando aquellos puntos que se consideran de mayor importancia o que requieren de una contextualización previa.

## 3.1. Sistemas multiagente tradicionales

Esta sección presenta los criterios de evaluación aplicables a cualquier tipo de sistema multiagente. Dichos criterios se han clasificado siguiendo la jerarquía que se presenta en la Figura 3.1. Cada una de estas categorías está desarrollada en las secciones posteriores.

Cabe resaltar que la categoría "Aspectos económicos" es aplicable tanto a metodologías, como a las herramientas de modelado e implementación.

### 3.1.1. Metodología y lenguaje de modelado

Esta sección define el proceso de evaluación de metodologías y lenguajes de modelado de sistemas multiagente. El cuestionario que se presenta permitirá comparar puntos de vista, ventajas, inconvenientes, etc.

El objetivo de una metodología es guiar en el proceso de desarrollo de una aplicación. Para ello las metodologías definen secuencias de procesos y guías que ayudan al desarrollador.

La mayoría de las metodologías de agentes se basan en las metodologías definidas en la ingeniería del software, tienen muchas características comunes, pero deben añadir las características y conceptos propios de los sistemas multiagente.

Teniendo en cuenta estas consideraciones, los criterios de evaluación se han dividido en 5 categorías: (1) Conceptos y propiedades de los sistemas multiagente; (2) Criterios relativos al modelado; (3) Criterios relativos al proceso de desarrollo; (4) Criterios relativos al uso de la metodología y el lenguaje de modelado; (5) Criterios relativos a cómo soportan ciertas características de alto nivel de abstracción. Esta clasificación está basada en trabajos de análisis y evaluación anteriores [Wooldridge and Ciancarini, 2001; Tran and Low, 2005; Lin et al., 2007].

#### Conceptos y propiedades

En esta sección se analiza si la metodología tiene en cuenta o no las características propias de los agentes y de los sistemas multiagente. A pesar

de la confusión en la definición de qué es un agente y un sistema multiagente, hay características comúnmente aceptadas y usadas [Wooldridge and Ciancarini, 2001; Bordini, Dastani, and Winikoff, 2007; Lin et al., 2007].

Estas características se han agrupado en *Características básicas* que abordan las propiedades más importantes de los sistemas multiagente y las *Características avanzadas* que abordan características más complejas y abstractas que pueden ser deseables para cierto tipo de sistemas multiagente pero que no son necesarias en la mayoría de ellos.

CARACTERÍSTICAS BÁSICAS

| |
|---|
| **Agent architecture:** Is the methodology or the modeling language focused on a specific agent architecture?<br>O No O Yes on _____ |
| **Platform dependency:** Is the development phase of the methodology focused on a specific deployment platform?<br>O No O Yes on _____ |
| **Architecture dependence:** Is the development phase of the methodology focused on a specific agent architecture?<br>O No O Yes on _____ |
| **Autonomy:** Can the models support and represent the autonomous feature of agents? Which technology is used to represent this?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable<br>Technology used: _____ |
| **Reactivity:** Can the models support and represent the agent's ability to respond in a timely manner to changes in the environment?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Proactiveness:** Can the models support and represent the agent's ability to pursue goals over time?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Cooperative behaviour:** Can the models support and represent the ability to work together with other agents to achieve a common goal?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Communication ability:** Can the models support and represent the ability to communicate with other agents?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Communication language:** The communication language used by the agents is based on:<br>O Signals (i.e low level languages) O Speech acts O Other, please specify _____ |

CARACTERÍSTICAS AVANZADAS

| |
|---|
| **Non-cooperative agents:** Does the methodology take into account non-cooperative or self interested agents?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Mental attitudes:** Can the models support and represent the use of agent mental attitudes like beliefs, desires and intentions?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable<br>Which mental attitudes are supported?: _____ |
| **Adaptability:** Can the models support and represent the ability of the agents to learn and improve with experience?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable<br>Technology used: _____ |
| **Temporal continuity:** Can the models support and represent temporal continuity of agents?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Deliberative capability:** Can the models support and represent the capability of the agent to select some possible plans to solve a problem and deliberate to choose the most appropriate one in each situation?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Inferential capability:** Can the models support and represent the ability to act on abstract task specifications?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Meta-management:** Can the models support and represent the ability of an agent to reason about a model of itself and of other agents modeled?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |

### Modelado

La notación y las técnicas de modelado son la clave para representar y modelar los elementos del sistema y sus actividades.

Estos criterios analizan cómo las técnicas que presentan las metodologías permiten el modelado de ciertos aspectos fundamentales en el desarrollo de sistemas multiagente, es decir, cómo la metodología es soportada por el lenguaje de modelado y si dicho lenguaje es capaz de representar los agentes, su interacción y la interacción con su entorno.

A su vez, consideran aspectos puramente de la ingeniería del software pero que también afectan al desarrollo de estos sistemas, tales como la expresividad y modularidad de los modelos.

| |
|---|
| **Modeling language representation:** Which kind of representation is used?<br>O Formal O Informal O Mixed |
| **Metamodels:** Is the methodology based on metamodels?<br>O No O Yes |
| **Kind of models:** Which models are used? Please add a brief description of each one. |
| **Model functionality:** Which of your models represents the following functionalities? If this functionality is not represented please write "Not supported".<br>O Roles, abilities, capabilities:_____<br>O Functionality:_____<br>O Interaction between agents:_____<br>O Interaction with the environment:_____<br>O Agent features:_____<br>O Social relationships:_____ |
| **Models dependence:** Is there a high dependence between the models?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Complete notation:** Can the modeling language support and represent all the concepts expressed by the methodology?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which can not be expressed? _____ |
| **Concurrency:** Can the models support and represent concurrent processes and the synchronization of concurrent processes?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Clarity:** Is the number of concepts expressed in a single diagram manageable?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Completeness:** Can the models support and represent all of the necessary concepts that describe the associated methodologies?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Protocols:** Can the models support and represent protocols, i.e., the definition of the allowable conversations in terms of the valid sequences of messages?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |
| **Different levels of abstraction:** Does the methodology and its models provide support for modeling at various levels of abstraction and detail?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Human Computer Interaction:** Does the methodology and its models provide support for modeling user interface and system-user interaction?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

| |
|---|
| **Modularity:** Does the methodology and its models provide support for modularity of design components?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Extensible:** Is the modeling language extensible?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Environment:** Does the modeling language provide support for modeling the environment of the agents and the agent effectors and perceptors?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Dynamic environment:** Does the modeling language provide support for modeling environment changes?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Dynamic roles:** Does the modeling language provide support for model changes in what abilities each agent has at each moment?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Resources:** Does the modeling language provide support for model agent external and internal resources, and their restrictions?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **External systems:** Does the modeling language provide support for describing the interaction with external systems?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

### Proceso de desarrollo

Toda metodología debe proporcionar un completo proceso de desarrollo. Un proceso de desarrolllo es una secuencia de pasos que guían a los usuarios a construir un sistema software durante diferentes etapas del ciclo de vida de la aplicación.

Hay metodologías que cubren de forma completa todo el proceso de desarrollo, es decir, desde la extracción de requisitos hasta la implementación final [Pavon, Gomez-Sanz, and Fuentes, 2005]. A pesar de eso, la mayoría se limitan a abordar la etapa de análisis y diseño [Giret, 2005; Argente, 2008].

Es importante analizar cómo cada metodología da soporte al proceso de desarrollo, qué etapas cubre y qué guías ofrece en cada una de ellas.

| |
|---|
| **Development lifecycle:** Which software-development process follows the methodology?<br>O Iterative O Waterfall O Others _____ |

**Coverage of the lifecycle:** What phases of the lifecycle are covered by the methodology? What is the level of support of each phase?
O Extraction of requirements: O None O Low O Medium O High O Don't know O Not Applicable
O Analysis: O None O Low O Medium O High O Don't know O Not Applicable
O Design: O None O Low O Medium O High O Don't know O Not Applicable
O Implementation: O None O Low O Medium O High O Don't know O Not Applicable

**Guidelines:** Do the methodology and the modeling language provide guidelines for the following issues?

- *Consistency guidelines:* Are there guidelines and techniques for consistency checking both within and between models?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

- *Consistency between levels of abstractions:* Are there guidelines for ensuring consistency between levels of abstractions within each model and between different models?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

- *Estimating guidelines:* Are there guidelines for estimating the number of agents required, the implementation time or other useful features?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

  Which _____

- *Support for decisions:* Are there guidelines on when to move between phases?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

- *Model derivation:* Are there guidelines for transforming models into other models, or partially create a model from the information present in another?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

- *Support for verification and validation:* Does the methodology contain rules to allow for the verification and validation of the correctness of developed models and specifications?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

**Development approach:** Which development approach is supported?
O Top-down approach O Bottom-up approach O Both O Indeterminate

**Approach towards MAS development:** Is the methodology OO-based or knowledge-engineering based?
O OO-based O Knowledge-engineering based O Other _____

**Application domain:** Is the methodology applicable to a specific application domain?
O No O Yes _____

| |
|---|
| **Model-central element:** What is the model-central element?<br>O Agents O Organizations O Services O Other _____ |
| **Interaction protocols:** Does the modeling language support the use of predefined interaction protocols?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Client communication:** Does the methodology provide support and facilitate communication between designers and users?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Models Reuse:** Does the methodology provide, or make it possible to use, a library of reusable models?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

### Uso e ingeniería del software

En esta sección se analizan criterios prácticos relativos a la ingeniería del software que evalúan la ejecución de los diferentes pasos del proceso de desarrollo propuesto por la metodología y el desarrollo de sus modelos.

| |
|---|
| **Unambiguity:** Does the notation of the modeling language have unambiguous mapping of concepts to symbols, uniform mapping rules and no overloading of notation elements?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Preciseness of models:** Is the mapping between notation and semantics clearly defined?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Expressiveness:** Are the models capable of capturing each concept at a high level of detail?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Consistency checking:** Does the model representation allow for consistency checking between their elements?<br>O No O Yes |
| **Notation simplicity:** Is notation semantically and syntactically simple across models?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Facility to understand:** Are the models and process steps easy to understand?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Facility to learn:** Are the notation and process steps easy to learn?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Facility to use:** Are the notation and process steps easy to use?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Refinement:** Does the methodology and its models provide support for a refinement-based design approach?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

| |
|---|
| **Documentation:** Are the use and features of the tool well documented?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Examples** Are there any complete examples of the use of the tool available?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

### Soporte a arquitecturas de alto nivel

En esta sección se analiza si la metodología soporta el diseño de arquitecturas de alto nivel tales como sistemas abiertos, agentes móviles, etc.

| |
|---|
| **Open systems:** Do the methodology and the modeling language provide support for open systems with heterogeneous agents?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Mobile agents:** Do the methodology and the modeling language provide support for mobile agents?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Security:** Do the methodology and the modeling language provide support for security techniques in agent applications?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Scalability:** What size of MAS is the methodology suited for?<br>O Small O Medium O Large O All |
| **Support for mobile agents:** Do the methodology and the modeling language support the use of mobile agents in MAS?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Support for ontology:** Do the methodology and the modeling language support the use/integration of ontology in MAS?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Support for MAS organizations:** Do the methodology and the modeling language support the use of organizational MAS?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Support for the integration with web services:** Do the methodology and the modeling language support integration with web services?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

### 3.1.2. Herramientas de desarrollo

Las metodologías proporcionan guías para ayudar a los desarrolladores durante el proceso de desarrollo, pero las metodologías son simplemente especificaciones teóricas y son necesarias herramientas de desarrollo que den soporte a dichas metodologías y permitan tanto el modelado como la implementación de los sistemas multiagente.

Como se muestra en la Figura 3.1, esta sección se divide en 5 categorías. Las dos categorías principales son *Herramienta de modelado* que analiza las técnicas que se utilizan para modelar este tipo de sistemas y *Herramienta de implementación* que se encarga de analizar las técnicas y funcionalidades ofrecidas por el entorno de implementación.

La categoría *Métodos-Herramienta de modelado* se encarga de analizar la distancia entre lo que especifica teóricamente la metodología y el lenguaje de modelado y lo que finalmente se puede modelar con la herramienta.

La categoría *Modelado-Implementación* se encarga de analizar la distancia entre lo que se ha modelado y la implementación final.

Finalmente, la categoría *Características técnicas* sirve para analizar desde un punto de vista técnico y del usuario final tanto la herramienta de modelado como de implementación.

#### Herramienta de modelado

Las herramientas de modelado permiten la transformación de las ideas y los conceptos abstractos de la metodología en diagramas y modelos usando un lenguaje específico de modelado.

Algunas metodologías, como [Arcos et al., 2005; Pavon, Gomez-Sanz, and Fuentes, 2005], ofrecen su propio entorno de modelado cubriendo así las características específicas de estas metodologías. Por el contrario, otras metodologías sólo ofrecen guías teóricas y se ha de utilizar herramientas de modelado de la ingeniería del software para realizar el modelado.

Esta sección analiza las características y la funcionalidad que deben ofrecer estas herramientas para facilitar la tarea de modelado.

**Kind of representation supported:** What kind of representation is used?
O Graphics O Formal languages O Mixed

**Automated generation of parts of the models:** Does the modeling tool automatically generate parts of the models?
O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

**Automated generation of models from requirements:** Is the modeling tool able to read a definition of the requirements and generate part of the models?
O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

**Store model language:** Does the modeling tool store the models in a standard language?
O No O Yes, they are stored using: _____

**Support for model checking:** Does the modeling tool include the support and tools to apply model checking?
O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree

**Support for ontology:** How does the modeling tool support the definition of the ontology?
O No support O Allows its implementation O Allows it to be imported from other programs Which are: _____

**Static verification tools:** Does the tool provide a static verification?

- *Detect inconsistences:* Does the modeling tool detect inconsistences within and between models?
  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree
- *Detect incompleteness:* Does the modeling tool detect incompleteness within and between models?
  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree
- *Propose solutions:* Does the modeling tool propose solutions when it detects an error within or between models?
  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree
- *Others:* Does the modeling tool offer other static verification tools?
  O No O Yes, which are: _____

**Dynamic verification tools:** Does the tool test the behaviour of the applications using simulations, i.e. simplified system prototypes?
O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree
Techniques used: _____

### Métodos-Herramienta de modelado

En esta sección se analiza la distancia entre lo que define la metodolgía y lo que puede ser modelado por la herramienta. Esta distancia debe ser lo más pequeña posible, ya que de lo contrario se podrían producir inconsistencias y se complicaría el trabajo del diseñador que siguiera dicha metodología [Dastani et al., 2004; Sudeikat et al., 2005].

| |
|---|
| **Support for the modeling language:** Does the modeling tool support all of the features and concepts that define the modeling language?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are not supported? _____ |
| **Support for the kind of representation:** Does the modeling tool support all types of representation used by the modeling language?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are not supported? _____ |
| **Lifecycle coverage:** Does the modeling tool support all the development stages supported by the methodology?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are not supported? _____ |

**Development guidelines:** Does the modeling tool integrate the methodology development guidelines?

- *Consistency guidelines:* Are the guidelines and techniques for consistency checking both within and between models integrated in the modeling tool?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree O Not supported by the methodology

- *Consistency between levels of abstractions:* Are the guidelines for ensuring consistency between levels of abstractions within each model and between different models integrated in the modeling tool?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree O Not supported by the methodology

- *Estimating guidelines:* Are the guidelines for estimating the number of agents required, the implementation time or other useful features integrated in the modeling tool?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree O This feature is not supported by the methodology

  Which _____

- *Support for decisions:* Are the guidelines regarding when to move between phases integrated in the modeling tool?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree O Not supported by the methodology

- *Model derivation:* Are the guidelines for transforming models into other models, or partially creating a model from the information present in another, integrated in the modeling tool?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree O Not supported by the methodology

- *Support for verification and validation:* Are the rules that allow the verification and validation of the correctness of developed models and specifications integrated in the modeling tool?

  O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree O Not supported by the methodology

### Herramienta de implementación

Las herramientas de implementación permiten transformar el diseño del sistema en el código final. En esta sección se analiza tanto el soporte específico que ofrece la herramienta a la implementación de sistemas multiagente, como las características típicas de este tipo de herramientas en la ingeniería del software.

| |
|---|
| **Platform dependent:** Does the tool allow the implementation of code for any MAS-execution platforms?<br>O Yes O No, only for: _____ |
| **Debugging facilities** Does the tool offer debugging facilities?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Generation of graphical interfaces:** Does the tool make it possible to generate graphical interfaces?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Limited systems:** Does the tool support the development of systems with some limitations, i.e., the development of systems that are going to be executed in limited devices like mobile phones?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Real time control:** Does the tool provide facilities to use real time control techniques?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Security issues:** Does the tool provide facilities to include security issues in the code of the MAS?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Physical environment models:** Does the tool provide a library of physical parts simulators in some kinds of systems for testing the functionality and correctness of the developed system?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Utility agents:** There are different agents offering services that do not depend on the particular application domain. Does the implementing tool provide the code of some utility agents?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

| |
|---|
| **Reengineering:** Does the tool make it possible to use reengineering techniques?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

### Modelado-Implementación

Esta sección analiza la distancia entre lo que se diseña y modela y lo que finalmente se puede implementar. Como se explica en [Xue, Zeng, and Liding, 2006; Bordini, Dastani, and Winikoff, 2007; Sudeikat et al., 2005], este es uno de los grandes problemas de los sistemas multiagente.

Muchas veces se diseñan sistemas utilizando técnicas de gran nivel de abstracción y en la mayoría de los casos la implementación se realiza de forma manual por los programadores.

Actualmente, se está trabajando en la transformación de código automática entre los modelos y el código final [Kostic, 2006; Gomez-Sanz and Pavon, 2005], pero este continúa siendo un campo de investigación abierto.

| |
|---|
| **Match MAS abstractions with implementation elements:** Do all of the concepts modeled have a direct translation into an implementation element?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which cannot be directly translated? _____ |
| **Automatic code generation:** Can the implementing tool read some models and generate parts of the code automatically?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which automatic code generation technology is used? _____ |
| **Specific platform facilities:** Does the implementing tool provide facilities for developing the code for a specific platform?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>For which platform/s? _____<br>What facilities are offered? _____ |

### Características técnicas

En esta sección se analizan características técnicas propias de la ingeniería del software tradicional, tales como qué requisitos son necesarios para instalar, ejecutar y usar una herramienta determinada.

Estos criterios pueden ser usados para evaluar tanto herramientas de

modelado como herramientas de implementación.

| |
|---|
| **Programming language:** With which programming languages has the tool been implemented?<br>O Java O C++ O C O Delphi O Other:_____ |
| **Installation requirements:**<br>▪ On which platforms can it be executed?<br>O PC O Mac O Mobiles O Others: _____<br>▪ Over which operating systems can it be executed?<br>O Windows, version: _____ O Linux, version _____ O Others: _____<br>▪ Is it light-weight?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Required expertise** Is it necessary to be an expert modeler and developer to use the tool?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Facility to learn:** Is it easy to learn to use the tool?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Facility to use:** Is the tool easy to use?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Extensibility:** Is it easy to add new functionalities to the tool?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Scalability:** Is the tool ready to develop applications any scale (small systems or large-scale applications)?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Collaborative development:** Does the tool provide support for collaborative development?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Documentation:** Are the use and features of the tool well documented?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Examples** Are any complete examples of the use of the tool available?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

### 3.1.3. Aspectos económicos

Los aspectos económicos son comunes tanto para las metodologías como para las herramientas de desarrollo.

Existe una tendencia a considerar que los aspectos económicos se reducen al coste de la aplicación, pero esto no es cierto ya que hay que tener en

cuenta otros aspectos muy importantes tales como la organización responsable de la herramienta, la documentación que ofrecen, etc.

En muchas ocasiones, la organización responsable da información sobre la fiabilidad y permanencia en el tiempo del producto que se está analizando.

| |
|---|
| **License:** Under which license is the evaluated method or tool is available?<br>O Open-source license: _____ O Private license |
| **Cost of the application:** What is the cost of a license for the application?<br>O Free O Not free, around: _____ |
| **Cost of its documentation:** What is the cost of its documentation?<br>O Free O Not free, around: _____ |
| **Vendor organization:** Which vendor organization is responsible for the evaluated method or tool?<br>O Industrial vendor: _____ O Academical vendor: _____ |
| **Updates:** Is the evaluated method or tool a static and definitive version or is it an open project with regular updates?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Technical service:** Does the vendor organization provide a technical service?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Examples of academic use:** How many academic applications have been developed using the evaluated method or tool?<br>O Any O One O 2-5 O 5-10 O More than 10 |
| **Examples of industrial use:** How many industrial applications have been developed using the evaluated method or tool?<br>O Any O One O 2-5 O 5-10 O More than 10 |

## 3.2. Sistemas multiagente organizacionales

Como se ha observado en el Capítulo 2, los sistemas multiagente organizacionales tienen necesidades y características que no se ajustan a los sistemas multiagente tradicionales. Por ello, esta sección se centra en las características que deben ser consideradas cuando se desarrollan este tipo de sistemas.

La categorización de dichos criterios es similar a la utilizada en los sistemas multiagente tradicionales, en primer lugar se analizan las necesidades a nivel de metodología; posteriormente la herramienta de modelado y de implementación y finalmente el lenguaje de implementación.

### 3.2.1. Metodología

La primera tabla resume los conceptos más generales que una metodología de agentes orientada a la organización debe cumplir.

El resto de los criterios hacen referencia a conceptos propios de las organizaciones y se han clasificado en 5 dimensiones: estructural, dinámica, funcional, normativa y de entorno. Dicha clasificación se basa en trabajos previos tales como [Coutinho, Sichman, and Boissier, 2005; Argente, Julian, and Botti, 2006].

| |
|---|
| **Model central-element:** Are organizations the model central-element?<br>O Yes O No |
| **Coverage of the lifecycle:** What phases of the lifecycle support organizational concepts and design? What is the level of support in each phase?<br>O Extraction of requirements: O None O Low O Medium O High O Don't know O Not Applicable<br>O Analysis: O None O Low O Medium O High O Don't know O Not Applicable<br>O Design: O None O Low O Medium O High O Don't know O Not Applicable<br>O Implementation: O None O Low O Medium O High O Don't know O Not Applicable |
| **Social patterns:** Does the methodology provide social patterns?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Modeling language representation:** What kind of representation is used to represent organizational concepts?<br>O Formal O Informal O Mixed |
| **Organizational models:** Has any organizational model been added? Please add a brief description of each one. |

| |
|---|
| **Complete notation:** Can the modeling language support and represent all of the concepts expressed by the methodology?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which can not be expressed? _____ |

DIMENSIÓN ESTRUCTURAL

Esta dimensión representa la estructura de la organización y todos los elementos que persisten en la organización independientemente de sus miembros. La estructura se define por sus roles, grupos, dependencias y sus enlaces.

| |
|---|
| **Topologies:** Which topologies are supported the methodology and by the modeling language?<br>O Flat-Structure O Pyramid Style O Chain of Values O matrix Structure-in-Five O Co-optation O Joint Venture O Other: _____ |
| **Topology guidelines:** Does the methodology provide guidelines to help developers to select the most appropriate topology for a specific application?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Other guidelines:** Does the methodology provide other guidelines to help developers to develop organizational systems?<br>O No O Yes:_____ |
| **Composed organization:** Do the methodology and the modeling language allow the creation of an organization inside another organization?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Social relationships:** Do the methodology and the modeling language allow the modeling of the type of social relationships between the agents and organizations?<br>O Knowledge links (who can get information about other agents) O Communication links (who can interact with it) O Authority links (who has control over others) O Other:_____ |
| **Role dependencies:** Do the methodology and the modeling language allow the modeling of role dependencies?<br>O Heritage O Communication O Compatibility O Coordination O Authority O Power control O Other:_____ |
| **Topology patterns:** Do the methodology and the modeling language provide patterns of social relationships or role dependencies based on a specific topology?<br>O No O Patterns of social relationships O Patterns of role dependencies O Both |
| **Meta-management based on norms:** Can the models support and represent the ability of an agent to reason about the norms of the organizations?<br>Level of support: O None O Low O Medium O High O Don't know O Not Applicable |

DIMENSIÓN DINÁMICA

Esta dimensión representa la evolución de la organización, es decir, cómo los agentes entran y salen de la organización y cómo cambian sus roles dependiendo de sus capacidades y objetivos de cada momento.

| |
|---|
| **Dymanical models:** Do the methodology and the modeling language allow the modeling of the dynamicity of the models? They are able to represent:<br>O The creation and destruction of the organizations O How agents go in/out of the organizations O How agents change their roles O The creation and destruction of new roles |
| **Context:** Do the methodology and the modeling language allow the context of the organizations to be modeled, i.e., the state of the organizations and agents during the execution?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Heterogeneous agents:** Do the methodology and the modeling language allow the interaction with heterogeneous agents to be modeled?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

DIMENSIÓN FUNCIONAL

Esta dimensión representa los objetivos de la organización y su descomposición en subobjetivos. También indica cómo se consiguen los objetivos, es decir, su descomposición en tareas y planes. Finalmente, define la funcionalidad que ofrece la organización y sus agentes.

| |
|---|
| **Goals:** Do the methodology and the modeling language allow the modeling of individual goals (agent goals) and global goals (organization goals)?<br>O Only individual goals O Only global goals O Both O None |
| **Global goal decomposition:** Do the methodology and the modeling language allow the modeling of the decomposition of global goals into specific individual goals?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Functionality:** Do the methodology and the modeling language allow the behaviour provided by the organizations to be specified?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

DIMENSIÓN NORMATIVA

Esta dimensión representa un conjunto de normas que controlan la organización. Las normas facilitan el mecanismo para conducir el comportamiento de los agentes [Lopez, Luck, and d'Inverno, 2006].

Actualmente, existen dos tendencias dentro de las metodologías orientadas a la organización. Por una parte hay métodos cómo Ingenias [Pavon, Gomez-Sanz, and Fuentes, 2005], que detallan los roles, los grupos y las relaciones pero no especifican sus normas sociales. Por otra parte, otros métodos y herramientas, como las Instituciones Electrónicas [Esteva et al., 2001], se centran en definir normas sociales y especifican políticas de control para asegurar su cumplimiento.

| |
|---|
| **Social norms:** Do the methodology and the modeling language allow social norms to be specified and control policies to be explicitly defined in order to establish and reinforce them?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Dynamic social norms:** Do the methodology and the modeling language support dynamic social norms?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Kinds of norms:** Which kinds of norms are supported by the methodology and the modeling language?<br>O None O Deontic (Obligations, Permissions and Prohibitions) O Legislatives (for creating, modifying or revoking norms) O Reinforcement (for controlling and penalizing) O Rewards O Other: _____ |
| **Temporal norms:** Do the methodology and the modeling language support norms with temporal restrictions?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Application level:** Which norm application levels support the methodology and the modeling language?<br>O Internal norms of the agents O Norms of a specific interaction O Organization norms O Other:_____ |
| **Inconsistent states:** Do the methodology and the modeling language the modeling of what is to be done when there is an inconsistent state (for example, when the norms forbid getting a goal)?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Formal representation:** Are the norms represented using a formal language?<br>O Yes O No |

DIMENSIÓN DEL ENTORNO

Esta dimensión representa todos los elementos del entorno que interaccionan con la organización.

| |
|---|
| **Stakeholders:** Do the methodology and the modeling language allow the representation of who benefits from the organizational results?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Depends on:** Do the methodology and the modeling language allow the representation of who the organization depends on?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

| |
|---|
| **Resources:** Do the methodology and the modeling language allow the representation of the mechanism used by the organizations to use an specific resource?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Perceptors and effectors:** Do the methodology and the modeling language allow the ability of the organizations to perceive and act over their environment to be modeled?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

### 3.2.2. Herramienta de modelado

La herramienta de modelado debe ofrecer facilidades para usar el lenguaje de modelado y la metodología orientada a la organización seleccionada.

| |
|---|
| **Support for the modeling language:** Does the modeling tool support all of the organizational features and concepts that define the modeling language?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are not supported? _____ |
| **Lifecycle coverage:** Does the modeling tool support modeling organizational issues at all of the development stages supported by the methodology?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are not supported? _____ |
| **Norms:** Does the modeling tool support modeling norms?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>How are the norms represented? _____ |
| **Inconsistency in norms:** Does the modeling tool detect inconsistences in the norms?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Check norms:** Does the modeling tool detect that the model does not follow specific norms (for example, when a forbidden interaction is modeled)?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Inconsistency in goals:** Does the modeling tool detect when a global goal is not decomposed into specific tasks and plans?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Consistency between goals and norms:** Does the modeling tool detect when the norms of a model do not allow a goal of this model to be obtained?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

| |
|---|
| **Developing guidelines:** Does the modeling tool integrate the guidelines provided by the methodology for developing organizational MAS?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are integrated? _____ |

### 3.2.3. Herramienta de implementación y lenguaje de programación

Los sistemas multiagente organizacionales se definen con mayor nivel de abstracción. Esto hace que la distancia entre lo que se modela y lo que finalmente se implementa aumente si las herramientas de implementación y de ejecución no se adaptan a este tipo de sistemas. Por ello deben ofrecer facilidades para implementar organizaciones [Argente et al., 2004].

| |
|---|
| **Platform dependent:** Is the implementing tool focused on a specific execution platform?<br>O No O Yes:_____ |
| **Organization representation:** How are the organizations materialized?<br>O Developers do not define the organization structure, although the observer can see an emergent organization.<br>O The organization exists as a specified and formalized schema, made by a designer but agents do not known anything about it and do not reason about it.<br>O Each agent has an internal and local representation of cooperation patterns which it follows when deciding what to do.<br>O Agents have an explicit representation of the organization which has been defined. |
| **Control mechanism:** How is the satisfaction of the organizational constraints ensured?<br>O The control mechanism has to be implemented by the developer.<br>O It is provided by the execution platform.<br>O Other: _____ |
| **Description of the organization:** Does the implementing language allow the representation of a description of the organizations?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Modeling concepts support:** Does the implementing language allow all of the organizational concepts defined in the models to be represented?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

| |
|---|
| **Types of topologies:** Which types of topologies can be directly represented by the implementing language?<br>O Flat-Structure O Pyramid Style O Chain of Values O matrix Structure-in-Five O Co-optation O Joint Venture O Other: _____ |
| **Kinds of norms:** Which kinds of norms can be represented by the implementing language?<br>O None O Deontic (Obligations, Permissions and Prohibitions) O Legislatives (for creating, modifying or revoking norms) O Reinforcement (for controlling and penalizing) O Rewards O Other: _____ |
| **Completeness of the modeling language:** Is the modeling language complete?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>It allows modeling:<br>O Create, destroy and modify organization structures. O Add, query and delete the agents of an organization. O Send messages to a whole organization. O Other:_____ |

## 3.3. Sistemas multiagente orientados a servicios

Al igual que los sistemas multiagente organizacionales, los sistemas multiagente orientados a servicios tienen características especiales que implican nuevos requisitos en los métodos y herramientas para desarrollar este tipo de sistemas. En esta sección se van a analizar dichas características siguiendo una categorización igual a la usada en la sección anterior.

### Metodología

En este apartado se analiza cómo la metodología cubre los aspectos relacionados con los servicios y con la integración de estos con los agentes.

| |
|---|
| **Integration type:** How is the relationship between agents and services considered?<br>O There is no conceptual distinction between agents and services. O Agents and services can communicate in a bidirectional way. O Agents invoke services but not vice versa. O Services invoke agents but not vice versa. |
| **Integrated methodology:** Does the MAS methodology take into account the integration between agents and services?<br>O Yes O No, another methodology is used to define services. O No there is no methodology to define services. |
| **Coverage of the lifecycle:** What phases of the lifecycle support the integration between agents and services? What is the level of support in each phase?<br>O Extraction of requirements: O None O Low O Medium O High O Don't know O Not Applicable<br>O Analysis: O None O Low O Medium O High O Don't know O Not Applicable<br>O Design: O None O Low O Medium O High O Don't know O Not Applicable<br>O Implementation: O None O Low O Medium O High O Don't know O Not Applicable |
| **Business process:** Does the methodology specify how business process languages can be used by agents or services to construct and execute business processes?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Development guidelines:** Do the methodology guidelines take into account the integration between agents and services and help to develop these kinds of systems?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Agents or services:** Does the methodology allow the processing of a user behavior description of desired functionality and recommend the implementation of the behavior via a service or an agent?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |

| |
|---|
| **Services and norms:** Does the methodology allow services with social norms to be integrated (to restrict access to a service, for example)?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Complete modeling language:** Can the modeling language support and represent services and their integration with agents?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Relationship between roles and services:** Can the modeling language represent the fact that a role uses or offers a service?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Service models:** Is any model added to represent the integration between agents and services or to describe services issues? Please add a brief description of each one. |
| **Modeling language representation:** What kind of representation is used to model the integration with services?<br>O Formal O Informal O Mixed O None |
| **Unambiguity:** Can the modeling language represent all of the specific features of agents, services and their integration without ambiguity and clearly defining which functionality is offered by the agents and which is offered by the services?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Service descriptions:** What kind of representation is used to model service descriptions?<br>O Formal O Informal O Mixed O None |
| **Completeness of service descriptions:** What features are represented in a service description?<br>O Its functionality. O The way to invoke it. O Which entities provide it. O Which entities are allowed to use it. |
| **Semantics services:** Is the methodology and the modeling language able to represent semantic services?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Interaction protocols:** What is the modeling language able to represent?<br>O Service composition O Service orchestration O Service choreography O Service and agent composition O Service and agent orchestration O Service and agent choreography |
| **Publishing services:** Is the modeling language able to represent the publication and the discovery of services?<br>O Service advertisement O Service discovery |

### Herramienta de modelado

La herramienta de modelado debe soportar todas las características definidas por la metodología y ser capaz de representar los servicios y su

integración con los agentes.

| |
|---|
| **Integrated modeling tool:** Does the modeling tool allow the modeling of both agents and services?<br>O Yes O Only agents but it can be connected to other modeling tools to specify services O Only agents O Only services |
| **Support for the modeling language:** Does the modeling tool support all of the service features and concepts that define the modeling language?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are not supported? _____ |
| **Lifecycle coverage:** Does the modeling tool support the modeling of the integration between agents and services at all of the development stages supported by the methodology?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Service descriptions:** Does the modeling tool provide facilities for specifying service descriptions?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which standard is used to specify service descriptions?_____ |
| **Semantic service descriptions:** Does the modeling tool offer facilities to specify semantic service descriptions?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which standard is used to specify service descriptions?_____ |
| **Checking descriptions:** Does the modeling tool allow errors in the definition of the service descriptions to be detected?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Developing guidelines:** Does the modeling tool integrate the guidelines provided by the methodology for developing systems that integrate agents and services?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which are integrated? _____ |
| **Automated code generation:** Does the modeling tool generate code from the models?<br>O None O Compleat agents O Agent skeletons O Compleat services O Services descriptions O Services skeletons |

| |
|---|
| **Inverse reengineering:** Does the modeling tool generate models from code?<br>O None O Service descriptions O Compleat services O Interactions O Agents features O Compleat |

### Herramienta y lenguaje de implementación

Los agentes y los servicios suelen usar estándares de comunicación e interacción diferentes, por lo que es necesario que las herramientas de implementación ayuden en la integración intentando hacer las interacciones lo más transparentes posible. De igual manera, deben ofrecer facilidades para implementar los agentes y los servicios.

| |
|---|
| **Integrated implementation:** Do the implementing tool and the programming language allow the implementation of both agents and services in the same tool?<br>O Yes O No, it only allows agents to be implemented O No, it allows agent and service invocations to be implemented O Other:_____ |
| **Integration with a modeling tool:** Is the implementing tool integrated with a modeling tool?<br>O No O They are integrated in the same tool O The implementing tool can read the models generated by the modeling tool |
| **Automatic code generation:** Can the implementing tool automatically generate parts of the agents and services code?<br>O No O For agents O For services O For the integration between them |
| **Implementation standards facilities:** Does the implementing tool provide facilities for implementing and checking the correct usage of these standards?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Which facilities are provided? _____ |
| **Integration gateway:** Does the implementing tool or the execution platform provide a gateway to allow communication between agents and services?<br>O No, it has to be implemented by the developer<br>O It is provided by the execution platform<br>O The developer has to use an external application like:_____<br>O Other: _____<br>Between which standards can it translate? Agent standards: _____ Service standards:_____<br>How is it implemented?<br>O A centralized gateway O There is an intermediary agent for each agent or service O Other:_____<br>What kind of communication does the integration gateway allow?<br>O Agents and services can communicate in a bidirectional way. O Agents invoke services but not vice versa. O Services invoke agents but not vice versa. |

| |
|---|
| **Publishing and discovery services:** Do both the programming language and the execution platform provide a mechanism for publishing service functionality?<br>O No O It allows services to discover agent functionality O It allows agents to discover services O Both<br>Please, explain the mechanism used:_____ |
| **Service descriptions:** Can the implementing tool and the programming language provide facilities for defining and consulting service descriptions, i.e., for defining complete service specifications, which include behavioral rules in addition to static interfaces, operations, preconditions, post conditions, and constraints?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Open systems:** Can the implementing tool provide facilities for implementing and checking the correct usage of these standards?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree |
| **Service composition:** Can services be composed from other services and agents behaviours?<br>O Yes O No, services only can be composed from other services but not from agent behaviors. O No, services cannot be composed.<br>How is this composition specified? _____<br>How is this composition translated into executable code?_____ |
| **API:** Is an API which allows the implementation of all of the features mentioned above provided?<br>O Yes, for the specific platform: _____ O No |
| **Testing applications:** Do the implementing tools allow the applications generated to be tested?<br>O Strongly Disagree O Disagree O Neutral O Agree O Strongly Agree<br>Does the implementing tool provide the following features?<br>O A testing server O A simulation of a local client application O A simulation of an agent client O A simulation of a web service client |

## 3.4. Métrica

Una evaluación numérica ofrece una visión general y rápida de los resultados de la evaluación ya que permite comparar los resultados obtenidos por cada uno de los productos evaluados en cada una de las categorías observando una sola tabla.

La métrica que presentamos a continuación se deriva del estudio de los trabajos comentados en el capítulo 2 y sobre todo en [Bitting, Carter, and Ghorbani, 2003].

$$result = \frac{\sum(W \ . \ R)}{\sum(W \ . \ max(P))} \ . \ 100$$

**W** representa la importancia de cada criterio de evaluación. A cada criterio de evaluación, es decir, a cada pregunta del cuestionario se le asocia un peso.

- 0 - Indica que este criterio no es cuantificable, por ejemplo, el hecho de que se use un arquitectura de agente u otra o que se use un lenguaje de implementación u otro no indica que la herramienta sea mejor o peor y por lo tanto no se puede evaluar numéricamente.
- 1 - Indica que este criterio es deseable pero no necesario.
- 2 - Indica que este criterio no es necesario porque es muy útil y por lo tanto es conveniente que el producto a evaluar lo cumpla.
- 3 - Indica que es necesario o muy importante para el desarrollo de sistemas multiagente.

**R** representa la respuesta que el evaluador le ha dado a cierto criterio. Cada respuesta del formulario tiene asociado un valor. En la mayor parte de los casos consiste en una escala del tipo 0, 25, 50, 75, 100 % que indica cómo de bien cubre el producto evaluado el criterio a considerar.

**max(P)** representa el mejor valor posible para ese criterio en concreto.

En los casos en que la respuesta es multivaluada, es decir, se le pide al evaluador que seleccione de una lista qué características cumple la herramienta, el valor de la respuesta se calcula con la siguiente fórmula:

$$result = \frac{\sum(W \ . \ \sum(R))}{\sum(W \ . \ max(\sum(P)))} \ . \ 100$$

Donde $\sum(R)$ representa el sumatorio de todas las respuestas que han sido marcadas.

$max(\sum(P))$ representa el sumatorio de todas las respuestas que podrían ser marcadas.

Finalmente, se aplica el producto vectorial de los pesos de los criterios con el valor de las respuestas obteniéndose así el valor numérico de la evaluación.

Se obtendrá un valor para cada una de las categorías que se han explicado en los apartados anteriores. Finalmente estas categorías también pueden ser ponderadas y aplicando el producto vectorial se obtiene el resultado final. Cada uno de estos valores ofrece una visión global del grado de desarrollo de esa herramienta en la categoría analizada. Estos valores sirven para comparar diferentes aproximaciones de forma rápida y aportando un primer resultado que se ampliará con las respuestas cualitativas a cada uno de los criterios del cuestionario.

## 3.5. Conclusiones

La evaluación de herramientas de desarrollo de sistemas multiagente es un área de investigación en auge pero que actualmente tiene grandes carencias. En este capítulo se ha contribuido al estado del arte mediante un método de evaluación de sistemas multiagente que permite analizar metodologías, herramientas de modelado y herramientas de implementación de forma completa.

Se presenta un método de evaluación qualitativo mediante la definición de una serie de criterios especificados en forma de cuestionario. A su vez,

se ha ofrecido un método de evaluación quantitativo que permite comparar metodologías y herramientas de desarrollo de forma general y rápida.

Este método de evaluación se ha ampliado añadiendo las cuestiones necesarias para evaluar cómo las metodologías y las herramientas de desarrollo se adaptan al diseño y la implementación de sistemas multiagente organizacionales y de sistemas multiagente orientados a servicios.

# Capítulo 4

# Masev

Tras el estudio del estado del arte en la evaluación de técnicas y herramientas para desarrollar sistemas multiagente presentado en el capítulo 2, se concluye que no hay ninguna herramienta de evaluación que permita de forma sencilla y homogénea analizar y comparar los métodos y herramientas involucrados en el proceso de desarrollo.

En el capítulo anterior (capítulo 3) se ha presentado un cuestionario que permite la evaluación a través de un método común, concreto y homogéneo. Estas características hacen que la evaluación sea un proceso sencillo y favorece la comparación entre herramientas.

En este capítulo se presenta Masev[1] (Multiagent System software evaluation framework). Masev es un entorno de evaluación que permite analizar y comparar métodos y herramientas para desarrollar sistemas multiagente. Para ello, Masev resume las características más importantes en el desarrollo de sistemas multiagente implementando el cuestionario mostrado en el Capítulo 3.

[1] http://masev.gti-ia.dsic.upv.es/

## 4.1. Motivación

El principal objetivo de Masev es ofrecer un sistema de evaluación completo, homogéneo y sencillo de usar.

Masev pretende obtener el mayor número de evaluaciones posibles, y con ello conseguir una gran base de información para analizar el estado del arte en el desarrollo de sistemas multiagente. Por ello, Masev se implementa en inglés y con una interfaz online que facilita su difusión y uso.

Como se ha observado en el capítulo 2, existen muchos trabajos dedicados a la evaluación de las herramientas para el desarrollo de sistemas multiagente. En ese capítulo se resaltaron las aportaciones y carencias de cada uno de ellos, pero en este punto cabe resaltar que ninguno de ellos ha tenido continuidad en el tiempo.

Estos trabajos se limitan a evaluar en un momento concreto ciertas metodologías o herramientas, pero al cabo de poco tiempo, suelen quedar obsoletos. Esto se debe a que el área de los sistemas multiagente es un campo de investigación en constante desarrollo, y por lo tanto muchas de sus metodologías y herramientas están en constante cambio.

Masev, al ser una herramienta online, permite a los usuarios actualizar sus evaluaciones en cualquier momento, actualizándose a su vez las comparativas con las otras herramientas.

## 4.2. Estructura

En esta sección se va a explicar la estructura de la herramienta online Masev. Esta estructura está resumida en la Figura 4.1

- **Home:** Introduce brevemente Masev y su funcionalidad.
- **Overview:** Describe qué es Masev, sus objetivos y su contribución al estado del arte.
- **Results:** Muestra los resultados de las evaluaciones previas.

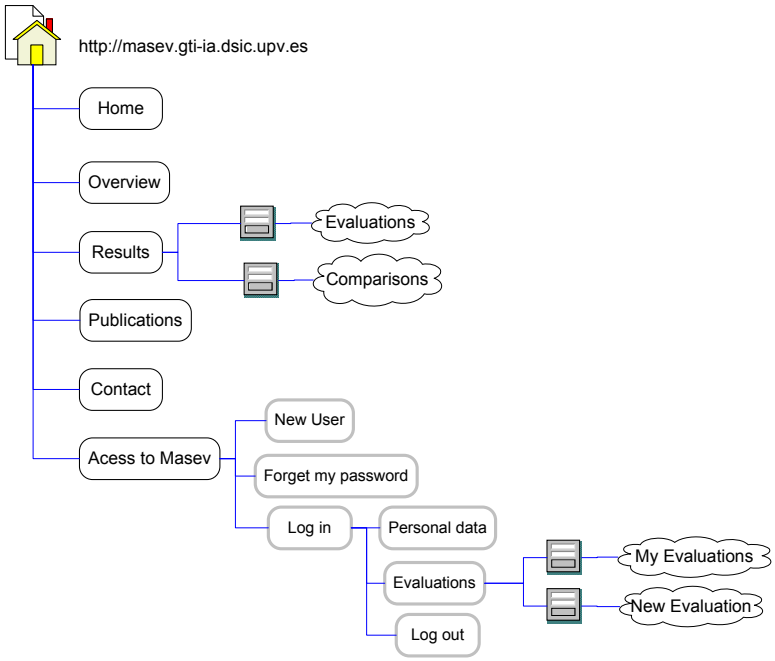


Figura 4.1: Estructura de Masev

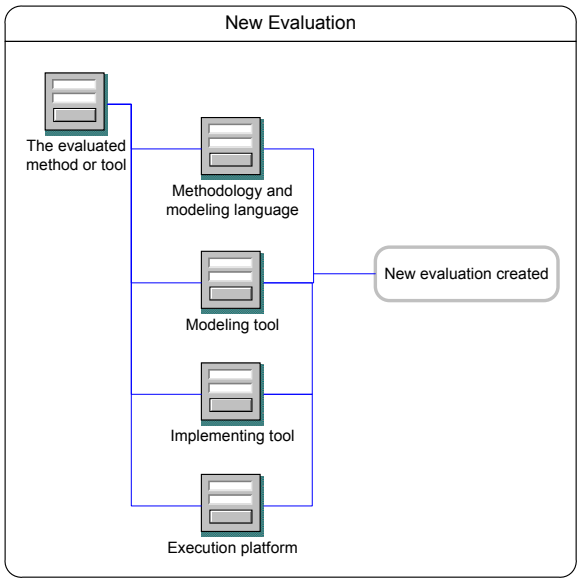


Figura 4.2: Estructura de New Masev Evaluation

- Muestra en una tabla comparativa todas las evaluaciones existentes en Masev sobre un método, herramienta o entorno de desarrollo en concreto. Para cada evaluación se remarca la experiencia del evaluador sobre la herramienta evaluada.
- Muestra una tabla comparativa de varias herramientas o métodos. El usuario puede seleccionar el tipo de evaluadores que quiere tener en cuenta (Creadores de la herramienta, usuarios, conocedores en detalle, usuarios familiarizados) y las herramientas o métodos que quiere comparar.

- **Publications:** Muestra las publicaciones sobre Masev y los criterios de evaluación seleccionados.

- **Contact:** Muestra información de contacto con los responsables de Masev.

- **Access to Masev:** Se le da la posibilidad al usuario de registrarse (New user), recordar su password a través de su email (Forget my password) o de poner su login y password y acceder a su página personal (Log in). Los usuarios son identificados por su email. En su página personal el usuario encuentra:

  - **Personal data:** La información que se almacena sobre él.
  - **Evaluations:** Muestra sus evaluaciones previas (My evaluations) con la posibilidad de actualizarlas o borrarlas en cualquier momento.

    A su vez, desde aquí tiene el acceso para crear nuevas evaluaciones (New evaluation). La estructura básica de este proceso se muestra en la Figura 4.2. En primer lugar se solicita información sobre qué tipo de herramienta se va a evaluar, el sistema admite la evaluación de metodologías y lenguajes de modelado, herramientas de modelado, herramientas de implementación, y entornos de desarrollo que engloben varias de estas herramientas.

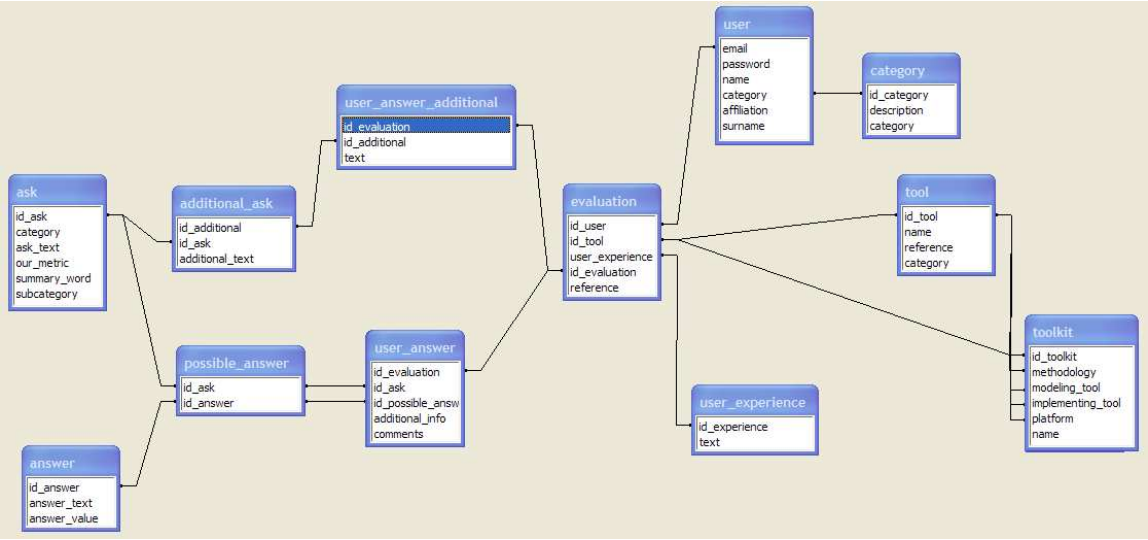


Figura 4.3: Estructura de la base de datos

A su vez, se indica la experiencia del evaluador y si se desea alguna referencia a la herramienta evaluada. A partir de este punto se van sucediendo una serie de formularios que implementan el cuestionario mostrado en el capítulo 2. Con el objetivo de minimizar el tiempo de evaluación, los formularios que se muestran dependen del tipo de herramienta a evaluar y de las respuestas dadas a preguntas previas, por ejemplo, si el usuario dice que la metodología no da soporte a organizaciones no se le mostrará el formulario que analiza este hecho.

- **Metric:** Muestra los valores numéricos obtenidos por las evaluaciones previas. El peso de los criterios de evaluación está preestablecido por el sistema, aunque el sistema está preparado para que sea el usuario mismo el que asigne los valores en función de sus necesidades.

- **Log out:** Cierra la sesión del usuario

La información recogida en las evaluaciones junto con las preguntas y la estructura del cuestionario se almacenan en una base de datos Mysql cuya estructura de tablas se muestra en la Figura 4.3.

## 4.3. Proceso de evaluación

Como requisito previo antes de acceder a la aplicación, los evaluadores deben registrarse en el sistema. El registro consta de la introducción de los datos que se pueden ver en la 4.4. Es un proceso muy sencillo tras el cual pueden acceder a la aplicación mediante su email y contraseña.

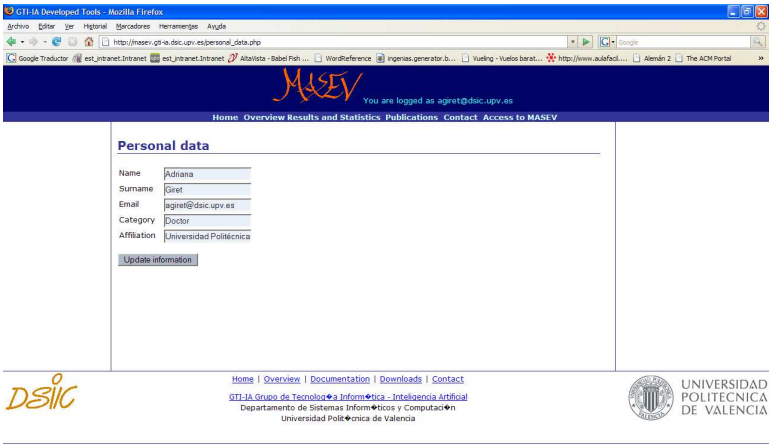


Figura 4.4: Información personal

Tal y como se ha explicado en la Sección 4.1, a través de su página personal cada usuario tiene acceso a sus evaluaciones anteriores y a la aplicación para introducir nuevas. En primer lugar la aplicación solicita información acerca de qué se va a evaluar y el nivel de experiencia del evaluador respecto al método o herramienta a evaluar.

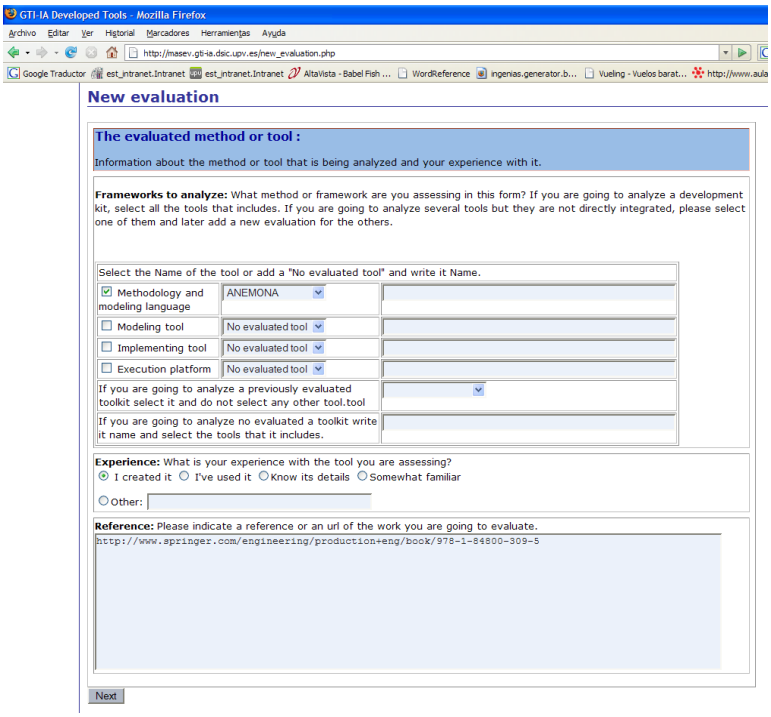


Figura 4.5: Formulario nueva evaluación

En función de qué tipo de sistema se va a evaluar se muestra la secuencia de formularios que implementan los criterios a utilizar. Cada uno de estos formularios corresponde a una categoría de las presentadas en el Capítulo 3. En el caso presentado en el ejemplo de la figura, lo que se evaluaba era una metodología. Por ello el usuario comenzó rellenando las cuestiones relativas a los conceptos y las propiedades de los sistemas multiagente (Figura 4.6), al modelado (Figura 4.7), etc. Cabe resaltar, que Masev intenta reducir el tiempo de evaluación al máximo por lo que sólo muestra las cuestiones relacionadas con el tipo de herramienta a evaluar. En la evaluación del ejemplo, el evaluador afirmó que dicha metodología daba cierto soporte a organizaciones, pero que no ofrecía soporte a servicios (Figura 4.8). Por ello, el sistema mostró las cuestiones relativas al soporte de organizaciones y ocultó las relativas a la integración de los agentes con los servicios.

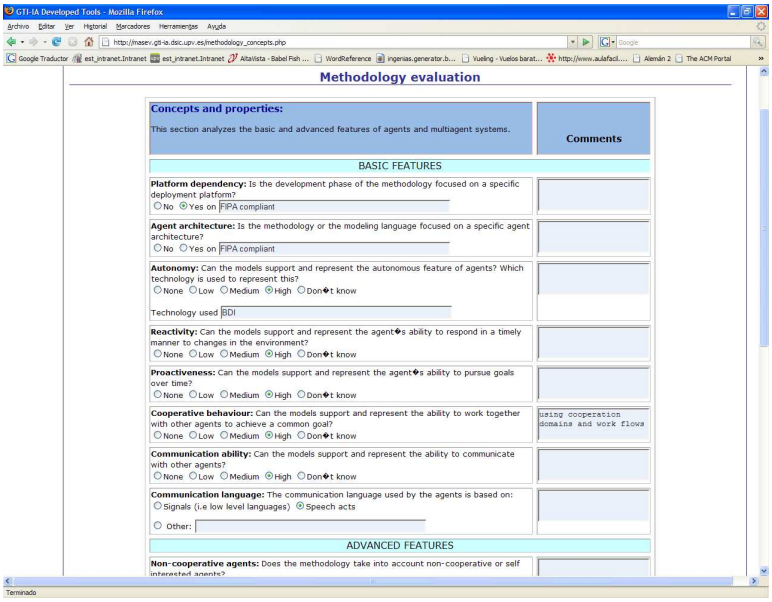


Figura 4.6: Formulario Concepts and properties

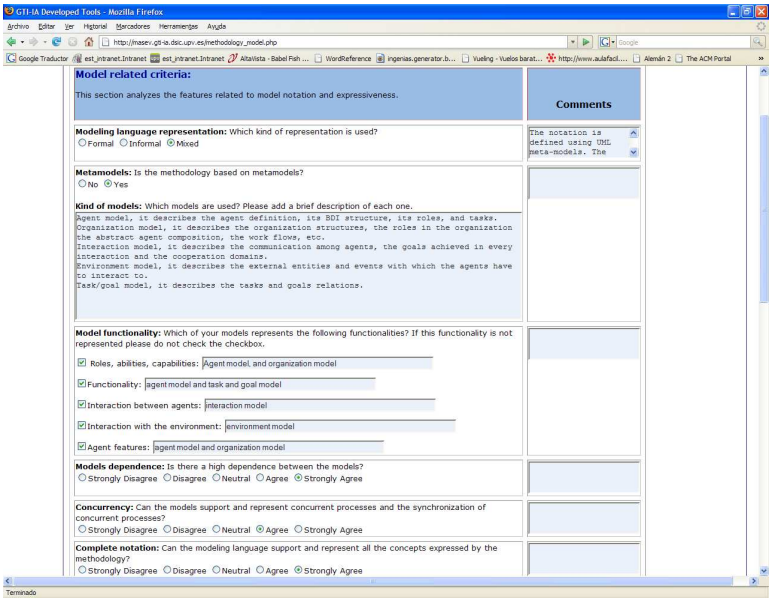


Figura 4.7: Formulario Model related criteria

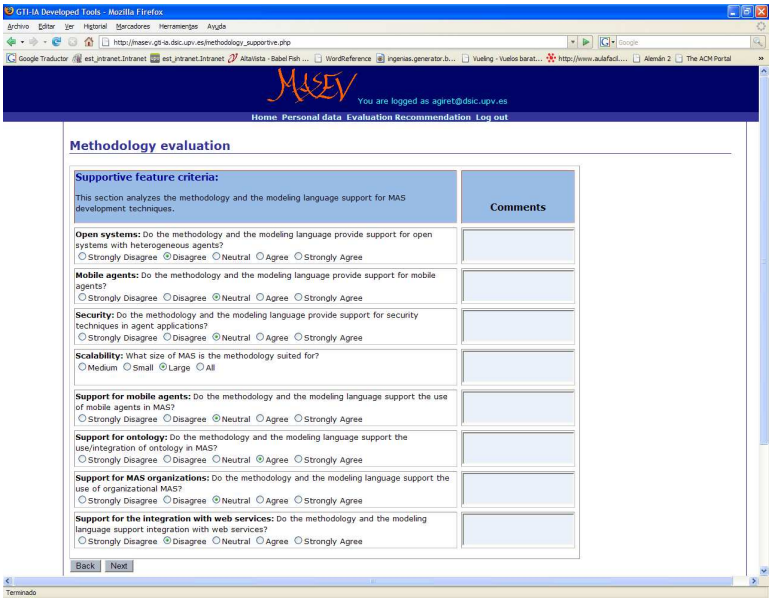


Figura 4.8: Formulario Supportive related criteria

Tras rellenar la secuencia de formularios presentada por Masev la evaluación finaliza y la información queda almacenada en la base de datos.

Para analizar los resultados Masev nos ofrece diferentes posibilidades. Se pueden comparar las diferentes evaluaciones de una misma herramienta o se pueden comparar diferentes herramientas del mismo tipo (Figura 4.9). En el ejemplo se pretendía comparar 4 metodologías distintas y por ello se optó por la segunda opción (Figura 4.10).

A su vez, Masev permite obtener valores numéricos a partir de la información obtenida tras las diferentes evaluaciones. Estos valores se derivan de la métrica detallada en la Sección 3.4. Para aplicar la fórmula el usuario tiene dos posibilidades o usar los pesos de los criterios de evaluación predefinidos por el sistema o definirse sus propios pesos y aplicarlos para calcular los valores finales.

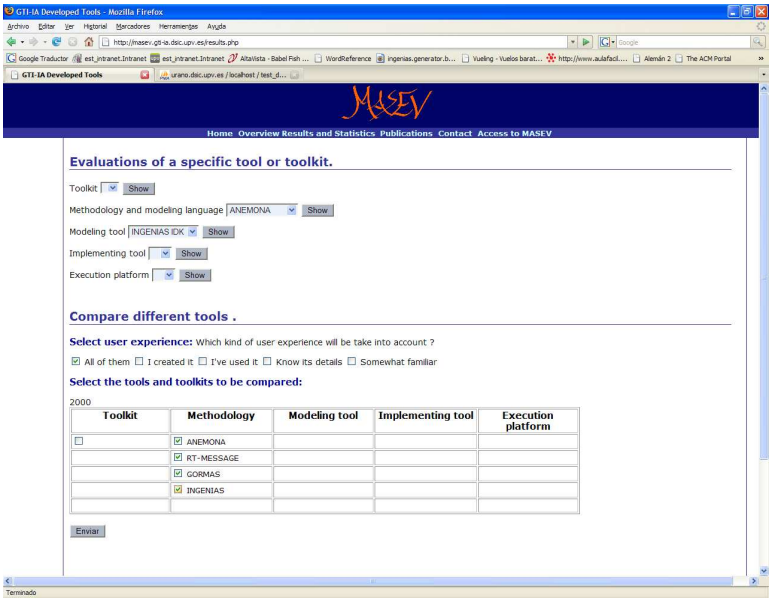


Figura 4.9: Selección del tipo de comparativa

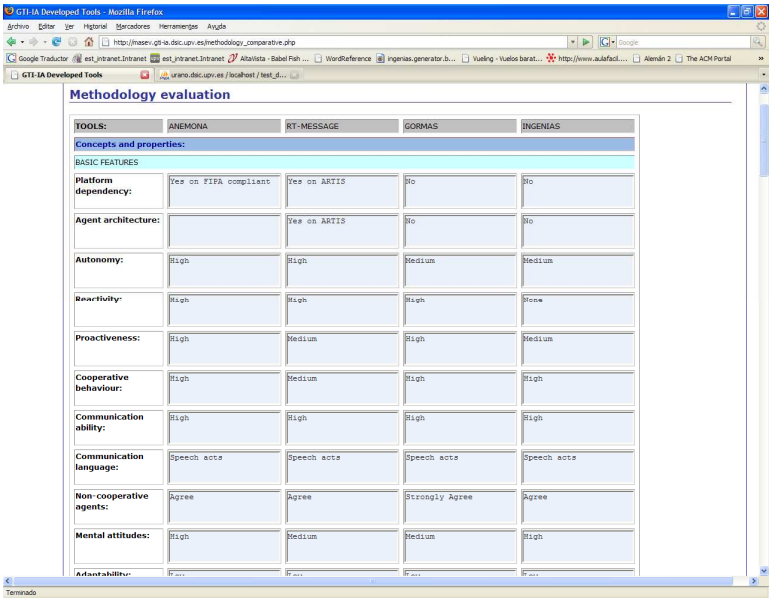


Figura 4.10: Estructura de Masev

## 4.4. Caso de estudio

Esta sección presenta el caso de estudio en el que se van a evaluar cuatro metodologías para desarrollar sistemas multiagente: RT-Message, Ingenias, Anemona y Gormas.

Este proceso se repitió para cada una de las metodologías, obteniéndose de este modo la base de conocimiento necesaria para comparar dichas metodologías.

El análisis de dichas metodologías con la herramienta Masev permitirá probar la utilidad, versatilidad y facilidad de uso del entorno de evaluación propuesto y de la herramienta Masev.

Esta sección se estructura en 3 partes. En primer lugar, se presentará una breve introducción a cada una de las metodologías. Posteriormente, se presentarán los resultados tanto cualitativos como cuantitativos de la evaluación y finalmente se resumirán las conclusiones más importantes.

#### RT-Message

RT-Message [Julian, 2002; Julian and Botti, 2004] está basado fundamentalmente en la metodología de desarrollo de sistemas multiagente conocida como MESSAGE y en el método RT-UML en lo que se refiere al desarrollo de sistemas de tiempo real. RT-MESSAGE es un conjunto de ideas flexibles y adaptables cubriendo las actividades de análisis, diseño e implementación.

Respecto al análisis amplía los modelos existentes en MESSAGE de tal forma que se pueden especificar comportamientos con restricciones temporales. En el diseño, se propone fundamentalmente el empleo de la arquitectura SIMBA [Carrascosa et al., 2003] para el modelado del sistema multiagente de tiempo real. Finalmente, en lo que se refiere a la implementación, a partir del diseño de los componentes necesarios, se plantea el uso de una herramienta visual de desarrollo de agentes ARTIS [Bajo et al., 2007], conocida como InSiDE [Julian et al., 2000]. Dicha herramienta permite, de forma gráfica, la implementación de todos los componentes de un agente de

este tipo, obteniendo como resultado un prototipo directamente ejecutable.

### Ingenias

Ingenias [Pavon, Gomez-Sanz, and Fuentes, 2005] es una metodología basada en la metodología Message que proporciona un conjunto de métodos y herramientas para el modelado de sistemas multiagente. La concepción de sistema multiagente en Ingenias es la de un conjunto de agentes inteligentes que cooperan entre sí para lograr satisfacer unos objetivos comunes.

El método de desarrollo de sistemas multiagente que propone Ingenias es la representación computacional de un conjunto de modelos, mostrando cada uno una visión parcial del sistemas multiagente. De este modo se definen los agentes que componen el sistema, las interacciones entre ellos, cómo se organizan, qué información del dominio es relevante describiendo a su vez cómo es el entorno y cómo se relaciona con el sistema. Con el objetivo de especificar los modelos, se definen metamodelos.

Los metamodelos de Ingenias derivan de los propuestos en Message, añadiendo: la integración de los metamodelos con las prácticas de ingeniería, un mayor nivel de detalle en los metamodelos, una mayor cohesión entre los metamodelos y representación del entorno del sistema. El resultado son 5 metamodelos que giran entorno a dos entidades, la organización y el agente.

Para plasmar cada uno de los modelos en diagramas, existe una herramienta gráfica, el Ingenias Development Kit (IDK) [Gomez-Sanz and Pavon, 2005] que facilita el modelado y la verificación de estos. A su vez el IDK también permite la generación automática de código de plantillas básicas en Java usando Jade.

### Anemona

Anemona [Botti and Giret, 2008; Giret, 2005]es una metodología multiagente para sistemas holónicos de fabricación. Esta metodología está basada en la noción de Agente Abstracto y en los requisitos de modelado de HMS. Anemona define un proceso de desarrollo mixto, y provee guías específicas

para HMS que ayudan al ingeniero del software a identificar e implementar holones.

El HMS se especifica dividiéndolo en aspectos más concretos que forman diferentes vistas del sistema. La forma en la que estas vistas están definidas se basa en la metodología Ingenias. Las extensiones que hemos hecho a los meta-modelos de Ingenias tratan con la noción de Agente Abstracto, la redefinición de relaciones para adaptarse a las entidades de modelado nuevas y la inclusión de características de modelado de tiempo real de la metodología RT-Message.

El proceso de desarrollo ofrece guías para HMS, y fases de desarrollo completas para el ciclo de vida del HMS. La primera fase Análisis de Requisitos del Sistema y la segunda Identificación y Especificación de Holones definen la etapa de análisis. El objetivo de la etapa de análisis es proveer una especificación de alto nivel del HMS a partir de los requisitos del sistema. La etapa de análisis adopta un enfoque descendente recursivo. El objetivo de la etapa de Implementación de Holones es producir el Código Ejecutable para la etapa de Instalación y Configuración. Finalmente las actividades de mantenimiento se llevan a cabo en la etapa de Operación y Mantenimiento.

### Gormas

Gormas [Argente, 2008; Argente, Julian, and Botti, 2008] propone un modelo de organización que permite describir los principales aspectos de las organizaciones: estructura, funcionalidad, normalización, dinamicidad y entorno. Este modelo consta de un conjunto de meta-modelos que extienden las propuestas de Ingenias y Anemona, empleando fundamentalmente los conceptos de unidad organizativa, servicio, norma y entorno. Además propone un conjunto de patrones de diseño, para facilitar el modelado de la estructura de la organización.

A su vez, Gormas ofrece una Guía Metodológica que permite el análisis y diseño de sistemas multiagente abiertos, desde la perspectiva de las organizaciones. La Guía Metodológica consta de un conjunto de fases que cubren el análisis de requisitos, el diseño de la estructura organizativa y

el diseño de la dinámica de la organización. Con estas fases se especifica, principalmente, cuáles son los servicios que ofrece la organización, cuál es su estructura interna y qué normas rigen su comportamiento.

### 4.4.1. Resultados

A continuación se muestran los resultados de la evaluación de las metodologías comentadas anteriormente. Dichos resultados se presentan en forma de tabla y se han obtenido utilizando la aplicación Masev. El motivo por el que se decidió que Masev presentara la información de esta manera es que gracias a esta estructura cualquiera podría buscar los criterios en los que está interesado y de forma rápida obtendría información del soporte que ofrece cada una de las herramientas evaluadas.

| | RT_Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|
| | Concepts | | | |
| **Platform dependency** | Yes on ARTIS | No | Yes on FIPA compliant | No |
| **Autonomy** | High | Medium | High | Medium |
| **Reactivity** | High | High | High | High |
| **Proactiveness** | Medium | Medium | High | High |
| **Cooperative behaviour** | Medium | High | High | High |
| **Communication ability** | High | High | High | High |
| **Communication language** | Speech acts | Speech acts | Speech acts | Speech acts |
| **Non-cooperative agents** | Agree | Agree | Agree | Strongly Agree |
| **Mental attitudes** | Medium | High | High | Medium |
| **Adaptability** | Low | Low | Low | Low |
| **Temporal continuity** | High | Medium | High | Medium |
| **Inferential capability** | High | Low | Medium | Medium |
| **Meta-management** | Medium | Low | Low | Medium |

Figura 4.11: Resultados categoría Concepts

De la Figura 4.11 se deriva que todas las metodologías tienen en cuenta las características básicas de los agentes. La única diferencia destacable a nivel conceptual es que así como Ingenias y Gormas son independientes de la plataforma final de ejecución, Anemona está diseñada sólo para plataformas que cumplan el estándard Fipa y RT-Message está diseñada para

la plataforma Artis. Esta es una distinción importante ya que un desarrollador que deba elegir entre estas metodologías, deberá elegir una u otra metodología en función de la plataforma de ejecución de agentes que vaya a utilizar.

| | RT_Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|
| | **Model** | | | |
| **Modeling language representation** | Mixed | Mixed | Mixed | Mixed |
| **Metamodels** | Yes | Yes | Yes | Yes |
| **Model functionality** | Roles, abilities, capabilities: agent, tasks/goals, organization<br><br>Functionality: tasks/goals<br><br>Interaction between agents: interaction<br><br>Interaction with the environment: agent, environment<br><br>Agent features: agent, organization | Roles, abilities, capabilities:agent model, organization model, task and goal model<br><br>Functionality: agent model, organization model, task and goal model<br><br>Interaction between agents: interaction model<br><br>Interaction with the environment: environment model<br><br>Agent features: agent model | Roles, abilities, capabilities: Agent model, and organization model<br><br>Functionality: agent model and task and goal model<br><br>Interaction between agents: interaction model<br><br>Interaction with the environment: environment model<br><br>Agent features: agent model and organization model | Roles, abilities, capabilities: organization, activity<br><br>Functionality: organization, activity<br><br>Interaction between agents: activity, interactions<br><br>Interaction with the environment: environment<br><br>Agent features: agent |
| **Models dependence** | Agree | Agree | Strongly Agree | Agree |
| **Concurrency** | Strongly Agree | Agree | Agree | Agree |
| **Complete notation** | Agree | Agree | Strongly Agree | Strongly Agree |
| **Clarity** | Agree | Agree | Strongly Agree | Agree |
| **Completeness** | Agree | Agree | Strongly Agree | Agree |
| **Protocols** | Strongly Agree | Agree | Strongly Agree | Strongly Agree |
| **Different levels of abstraction** | Neutral | Strongly Disagree | Strongly Agree | Strongly Agree |
| **Human Computer Interaction** | Disagree | Disagree | Neutral | Agree |
| **Modularity** | Agree | Neutral | Strongly Agree | Neutral |
| **Extensible** | Agree | Strongly Agree | Strongly Agree | Agree |
| **Environment** | Strongly Agree | Agree | Strongly Agree | Strongly Agree |
| **Dynamic environment** | Agree | Agree | Strongly Agree | Agree |
| **Dynamic roles** | Neutral | Disagree | Agree | Neutral |
| **Resources** | Strongly Agree | Agree | Strongly Agree | Strongly Agree |
| **External systems** | Agree | Agree | Strongly Agree | Agree |

Figura 4.12: Resultados categoría Model

Todos están basados en metamodelos y un lenguaje de modelado mixto (formal, informal). Todos ofrecen las funcionalidades básicas aunque las representan en diferentes metamodelos (Figura 4.12).

| | RT_Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|
| | **Process** | | | |
| **Development lifecycle** | Iterative | RUP | Recursive, iterative and incremental | Iterative |
| **Coverage of the lifecycle:** **Extraction of requirements:** | Medium | None | Medium | Médium |
| **Analysis:** | High | High | High | High |
| **Design:** | High | Medium | High | High |
| **Implementation:** | High | Medium | Medium | None |
| **Development approach** | Top-down approach | Indeterminate | Both | Top-down approach |
| **Approach towards MAS development** | OO-based | Knowledge-engineering based | Agent oriented | OO-based |
| **Application domain** | Yes | Yes | Yes | No |
| **Model-central element** | Agents | Agents | Abstract agent or holon | Organizations |
| **Interaction protocols** | Agree | Neutral | Disagree | Agree |
| **Consistency guidelines** | Neutral | Agree | Agree | Agree |
| **Estimating guidelines** | Agree | Disagree | Disagree | Disagree |
| **Support for decisions** | Agree | Agree | Strongly Agree | Agree |
| **Model derivation** | Agree | Agree | Strongly Agree | Agree |
| **Support for verification and validation** | Disagree | Neutral | Neutral | Disagree |
| **Client communication** | Disagree | Disagree | Strongly Agree | Disagree |
| **Models Reuse** | Agree | Disagree | Neutral | Agree |

Figura 4.13: Resultados categoría Process

Todas siguen un proceso de desarrollo estandarizado (Figura 4.13). Las etapas de análisis y diseño son abordadas por todas las aproximaciones. Aunque Ingenias no cubre la etapa de extracción de requisitos y Gormas no aborda la etapa de implementación.

A nivel de criterios prácticos relativos a la ingeniería del software son todas las aproximaciones similares (Figura 4.14), aunque la metodología Anemona ofrece unos modelos más fáciles de usar y con mayor precisión. Por el contraio, esta figura muestra diferencias considerables respecto a las arquitecturas a las que dan soporte cada una de las metodologías. Los agentes móviles sólo son considerados en Anemona y no de forma completa. Así como el modelado basado en organizaciones es soportado por todas las metodologías, sólo Gormas tiene en cuenta sistemas abiertos y la integración de los agentes con servicios.

| | RT_Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|
| | **Pragmatic** | | | |
| **Unambiguity** | Agree | Agree | Strongly Agree | Neutral |
| **Preciseness of models** | Agree | Agree | Strongly Agree | Neutral |
| **Expressiveness** | Agree | Agree | Strongly Agree | Neutral |
| **Consistency checking** | Disagree | Agree | Neutral | Agree |
| **Notation simplicity** | Agree | Agree | Strongly Agree | Neutral |
| **Facility to understand** | Neutral | Agree | Strongly Agree | Neutral |
| **Facility to learn** | Neutral | Agree | Strongly Agree | Neutral |
| **Facility to use** | Neutral | Neutral | Strongly Agree | Neutral |
| **Refinement** | Agree | Disagree | Strongly Agree | Agree |
| **Documentation** | Disagree | Agree | Strongly Agree | Agree |
| **Examples** | Neutral | Agree | Strongly Agree | Agree |
| | **Supportive** | | | |
| **Open systems** | Disagree | Strongly Disagree | Disagree | Agree |
| **Mobile agents** | Disagree | Disagree | Neutral | Disagree |
| **Security** | Disagree | Strongly Disagree | Neutral | Disagree |
| **Scalability** | Small | Large | Large | Large |
| **Support for mobile agents** | Disagree | Disagree | Neutral | Disagree |
| **Support for ontology** | Agree | Disagree | Agree | Neutral |
| **Support for MAS organizations** | Agree | Agree | Neutral | Strongly Agree |
| **Support for the integration with web services** | Disagree | Strongly Disagree | Disagree | Agree |

Figura 4.14: Resultados categorías Pragmatic y Supportive

| | RT_Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|
| **Model central-element** | No | No | No | Yes |
| **Coverage of the lifecycle:** **Extraction of requirements:** | High | None | Low | Medium |
| **Analysis:** | High | Medium | Medium | High |
| **Design:** | Médium | Medium | Medium | High |
| **Implementation:** | Low | Low | Low | Low |
| **Social patterns** | Neutral | Strongly Disagree | Disagree | Strongly Agree |
| | **Structural** | | | |
| **Topologies** | Flat-Structure Pyramid Style | Flat-Structure | Holonic | Flat-Structure Pyramid Style Chain of Values matrix Structure-in-Five Co-optation |
| **Topology guidelines** | Disagree | Strongly Disagree | Strongly Agree | Agree |
| **Other guidelines** | No | No | No | Yes |
| **Composed organization** | Disagree | Strongly Disagree | Strongly Agree | Agree |
| **Social relationships** | Communication links Authority links | Knowledge links Communication links Authority links | Knowledge links Communication links Authority links | Knowledge links Communication links Authority links |
| **Role dependencies** | Communication Coordination Authority | Communication Coordination Authority | Communication Coordination Authority | Heritage Coordination Authority |
| **Topology patterns** | No | No | Patterns of role dependencies | Patterns of social relationships |
| **Meta-management based on norms** | None | None | None | High |
| **Dynamical models** | --- | ---- | The creation and destruction of the organizations<br><br>How agents change their roles<br><br>The creation and destruction of new roles | The creation and destruction of the organizations<br><br>How agents go in/out of the organizations<br><br>How agents change their roles<br><br>The creation and destruction of new roles |
| | **Dynamic** | | | |
| **Context** | Disagree | Strongly Disagree | Disagree | Agree |
| **Heterogeneous agents** | Agree | Strongly Disagree | Disagree | Agree |
| | **Functional** | | | |
| **Goals** | Both | Both | Both | Both |
| **Global goal decomposition** | Agree | Agree | Strongly Agree | Strongly Agree |
| **Functionality** | Neutral | Agree | Strongly Agree | Agree |

Figura 4.15: Resultados categoría Organizational

| | RT_Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|
| | **Normative** | | | |
| **Social norms** | Disagree | Strongly Disagree | Disagree | Strongly Agree |
| **Dynamic social norms** | Disagree | Strongly Disagree | Disagree | Agree |
| **Kinds of norms:** | None | None | Deontic Legislatives Rewards | None |
| **Temporal norms** | Agree | Strongly Disagree | Strongly Disagree | Agree |
| **Application level** | Internal norms of the agents | None | None | Organization norms |
| **Inconsistent states** | Disagree | Strongly Disagree | Disagree | Neutral |
| **Formal representation** | No | No | No | Yes |
| | **Environment** | | | |
| **Stakeholders** | Agree | Strongly Disagree | Neutral | Strongly Agree |
| **Depends on** | Neutral | Strongly Disagree | Neutral | Agree |
| **Resources** | Neutral | Strongly Disagree | Strongly Agree | Strongly Agree |
| **Perceptors and effectors** | Neutral | Strongly Disagree | Strongly Agree | Strongly Agree |
| | **Modeling language** | | | |
| **Modeling language representation** | Informal | Informal | Mixed | Formal |
| **Organizational models** | None | there is an organizational model, but it is very simple | None | structural, functional, social, dynamic |
| **Complete notation** | Agree | Agree | Strongly Agree | Agree |

Figura 4.16: Resultados categoría Organizational

A pesar de que todas las aproximaciones ofrecen cierto soporte a organizaciones, se puede observar en la Figura 4.15 que cada una ofrece una serie de funcionalidades. Por ejemplo, a nivel estructural la metodología Gormas ofrece facilidades para diseñar muchas más estructuras topológicas que el resto y es la única que ofrece guías para seleccionar la estructura más adecuada dependiendo de los requisitos de la aplicación a implementar. A su vez, es la única que permite la definición de normas a nivel tanto interno de agentes, como de la organización a la que pertenezcan, como a nivel de las interacciones que se produzcan entre ellos (Figura 4.16).

El análisis de cómo las metodologías soportan la integración de los agentes con arquitecturas orientadas a servicios nos muestra que sólo Gormas ofrece cierto soporte. Gormas se centra en el análisis y el diseño y ofrece guías de desarrollo para estas etapas. El tipo de integración que plantea Gormas es bidireccional, es decir, que los agentes pueden invocar servicios

y los servicios pueden invocar a agentes.

| | Gormas |
|---|---|
| **Integration type** | Agents and services can communicate in a bidirectional way. |
| **Integrated methodology** | Yes |
| **Coverage of the lifecycle:** **Extraction of requirements:** | Low |
| **Analysis:** | Médium |
| **Design:** | Médium |
| **Implementation:** | Low |
| **Business process** | Neutral |
| **Development guidelines** | Agree |
| **Agents or services** | Neutral |
| **Services and norms** | Agree |
| | **Modeling language** |
| **Complete modeling language** | Agree |
| **Relationship between roles and services** | Agree |
| **Modeling language representation** | Formal |
| **Unambiguity** | Neutral |
| **Service descriptions** | Formal |
| **Completeness of service descriptions** | Its functionality.<br>Which entities provide it.<br>Which entities are allowed to use it. |
| **Semantics services** | Agree |
| **Interaction protocols** | Service composition<br>Service and agent composition |
| **Publishing services** | Service advertisement<br>Service discovery |

Figura 4.17: Resultados categoría Services

Por último, la Figura 4.18 muestra que ninguna metodología tiene licencia propietaria y que se han creado en entornos académicos. Anemona y RT-Message no tienen actualizaciones recientes, pero por el contrario Ingenias y Gormas están actualmente en constante desarrollo.

| | RT_Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|
| **Cost of the application** | Free | Free | Free | Free |
| **Cost of its documentation** | Free | Free | Free | Free |
| **Vendor organization** | Academical vendor: UPV | Academical vendor:GRASIA, UCM | Academical vendor: UPV | Academical vendor: UPV |
| **Updates** | Neutral | Strongly Agree | Neutral | Agree |
| **Technical service** | Neutral | Agree | Agree | Disagree |
| **Examples of academic use** | 2-5 | More than 10 | 2-5 | One |
| **Examples of industrial use** | One | 2-5 | 2-5 | Any |

Figura 4.18: Resultados categoría Economic

Los resultados numéricos mostrados en la Figura 4.19 se ajustan a la métrica comentada en el capítulo 3.4 y nos permite comparar de forma muy general las diferentes metodologías. Por ejemplo, viendo los resultados de la categoría "Supportive", es decir, la que analiza cómo soportan las metodologías cierto tipo de arquitecturas podemos observar que RT-Message e Ingenias tienen grandes limitaciones. De igual modo se observa que el proceso de desarrollo no está completamente especificado ni ofrece muchas facilidades en ninguna de las metodologías estudiadas, pero sobre todo debería ser mejorado en el caso de Ingenias. A nivel económico todas obtienen una buena puntuación debido a que todas son públicas y existen ejemplos de todas ellas.

| | | RT-Message | Ingenias | Anemona | Gormas |
|---|---|---|---|---|---|
| Methodology | Concepts (3) | 66,66 | 62,5 | 78,12 | 69,79 |
| | Model (3) | 78,04 | 66,46 | 93,29 | 80,48 |
| | Process (3) | 71,66 | 45,00 | 68,33 | 60,00 |
| | Pragmatic (2) | 57,40 | 69,44 | 96,30 | 59,26 |
| | Supportive (1) | 39,70 | 27,94 | 51,47 | 55,88 |
| | | | | | |
| Economical aspects (1) | | 75 | 90,63 | 81,25 | 70,31 |
| | | | | | |
| Total | | 67,59 | 59,95 | 80,35 | 67,35 |

| | | | | |
|---|---|---|---|---|
| Organizations | 41,38 | 19,39 | 46,55 | 71,98 |
| Services | 0,00 | 0,00 | 0,00 | 52,00 |

Figura 4.19: Resultados numéricos

## 4.5. Conclusiones

En este capítulo se ha presentado una herramienta online que implementa el entorno de evaluación presentado en el capítulo 3. Esta herramienta permite evaluar y comparar metodologías y herramientas de desarrollo de sistemas multiagente, analizando a su vez las características necesarias para desarrollar sistemas multiagente organizacionales y orientados a servicios.

Esta herramienta se ha probado satisfactoriamente analizando y comparando cuatro metodologías: RT-Message, Ingenias, Anemona y Gormas. A través de la experiencia obtenida en este caso de estudio se deriva que

Masev es un entorno de evaluación que permite evaluar herramientas de desarrollo de sistemas multiagente. El proceso de evaluación de cada una de estas metodologías llevó alrededor de 15 minutos y facilitó la información necesaria para analizar y comparar dichas metodologías. Por lo tanto, se puede concluir que Masev simplifica el proceso de evaluación reduciendo el tiempo necesario para obtener análisis completos de una herramienta. Ninguno de los evaluadores tuvo problemas al rellenar los diferentes cuestionarios y la información obtenida es fácilmente estructurada a través de Masev. De este hecho se puede derivar que el uso de cuestionarios reduce la complejidad de la evaluación y hace que la información suministrada no sea ambigua.

Las comparativas facilitadas por la herramienta han sido de gran utilidad para analizar las herramientas y detectar sus carencias. Masev estructura la información en forma de tablas por lo que es muy fácil encontrar similitudes y diferencias entre cada una de las herramientas comparadas. Finalmente, gracias a los resultados numéricos obtenidos tras la aplicación de la métrica hemos podido hacer un análisis y una comparación a nivel genérico de forma rápida y sencilla.

Al realizar el caso de estudio se observó que los resultados eran totalmente dependientes de la opinión del evaluador. Este hecho introduce demasiada subjetividad en el proceso, a pesar de que los evaluadores seleccionados eran expertos, incluso en algunos casos creadores de las metodologías estudiadas. Por ellos se preparó Masev para soportar múltiples evaluaciones de una misma herramienta o metodología y calcular el valor medio a la hora de mostrar las comparativas. A su vez, Masev permite al usuario filtrar en función de la experiencia del evaluador y por lo tanto sólo considerar las evaluaciones de cierto tipo de evaluadores.

# Capítulo 5

# Conclusiones y trabajo futuro

En este trabajo hemos presentado un entorno de evaluación para metodologías, herramientas y técnicas de desarrollo de sistema multiagente. A su vez, hemos añadido dos módulos que permiten la evaluación y el análisis de las características más importantes en el desarrollo de sistemas multiagente organizacionales y de sistemas multiagente orientados a servicios.

Dicho entorno de evaluación se basa en el análisis y comparación de ciertos criterios de evaluación a los que posteriormente se les aplica una métrica para obtener comparativas tanto cualitativas como cuantitativas. La métrica propuesta en este trabajo es una extensión de la propuesta por Bitting y Carter en [Bitting, Carter, and Ghorbani, 2003].

Con el objetivo de determinar cuáles son los criterios de evaluación más importantes hemos realizado un estudio del estado del arte en el desarrollo de sistemas multiagente tradicionales, orientados a la organización y orientados a servicios. También hemos analizado el estado del arte en la evaluación de métodos y técnicas de desarrollo de este tipo de sistemas. Finalmente, desarrollamos diferentes aplicaciones con el objetivo de analizar el estado del arte no sólo desde un punto de vista teórico sino también desde el punto de vista práctico del desarrollador.

A partir de estos estudios hemos determinado los criterios de evaluación más importantes estructurados en forma de cuestionario para facilitar su uso. Estos criterios permiten la evaluación de metodologías, lenguajes de

modelado, herramientas de modelado y herramientas de implementación, analizando a su vez la distancia entre lo definido por la metodología y lo que permite modelar la herramienta y la distancia entre los modelos realizados y la implementación final.

Por último, hemos desarrollado Masev que es una aplicación online que implementa el entorno de evaluación presentado. Tal y como se observa en el caso de estudio presentado, esta aplicación permite de forma rápida y sencilla evaluar y comparar métodos y herramientas de desarrollo de sistemas multiagente.

Por lo tanto, las contribuciones de este trabajo son las siguientes:

- Para los desarrolladores de métodos y herramientas de desarrollo de sistemas multiagente:
  - Un mecanismo para comparar sus sistemas con otros del mismo tipo
  - Un mecanismo para detectar carencias en sus herramientas
  - Un mecanismo para publicitar sus herramientas y de probar sus contribuciones respecto a las ya existentes.
  - Una lista de las características que deben tener las herramientas para desarrollar sistemas multiagentes tradicionales, orientados a organizaciones y orientados a servicios.
- Para los desarrolladores de sistemas multiagente
  - Un mecanismo para comparar diferentes métodos y herramientas, y por lo tanto, un sistema que ayuda a elegir la más adecuada en función de sus necesidades.
- Para la comunidad de sistemas multiagente
  - Un mecanismo que permite crear y actualizar un completo estado del arte en lo que respecta a métodos y herramientas de desarrollo de sistemas multiagente

- Un estudio sobre los nuevos requisitos que surgen al introducir en los sistemas multiagente el concepto de organización y al integrar los agentes con servicios.

Como trabajo futuro pretendemos ampliar el entorno de evaluación añadiendo la evaluación de plataformas de ejecución de agentes y ofreciendo un módulo de recomendación. Dicho módulo ofrecerá un ranking de las herramientas a utilizar más convenientes en función de los requisitos del sistema a desarrollar. Otra de las ampliaciones futuras de Masev es ofrecer al usuario la posibilidad de definir su propia métrica a través de la herramienta. De este modo, la información de las evaluaciones de Masev podrían servir para probar y definir nuevas métricas de evaluación.

A su vez, pretendemos publicitar la aplicación online Masev con el objetivo de obtener el mayor número posible de evaluaciones y conseguir así un estudio completo del estado del arte en el desarrollo de este tipo de sistemas.

Con esta información se obtendrán conclusiones acerca de las carencias y tendencias de este tipo de herramientas. Estas conclusiones guiarán los futuros desarrollos del grupo de investigación GTI-IA.

Este trabajo es el punto de partida para la obtención de un entorno de evaluación que englobe no sólo la evaluación informal a través de criterios sino también la evaluación formal tanto de las herramientas de evaluación como de los modelos y el código generado.

## 5.1. Publicaciones

- E. Garcia, S. Valero, E. Argente, A. Giret, and V. Julian.

  A FAST method to achieve Flexible Production Programming Systems.

  IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, 38(2):242 252, 2008.

  Índice de impacto JCR: 0.706

  Indexed in Core Conference Ranking (B)

- S. Valero, E. Garcia, E. Argente, A. Giret, and V. Julian.

  Flexible and Dynamic Production Programming Tool Based on MAS Technology.

  INFOCOMP: Journal of Computer Science, pages 1-8, 2007.

- Emilia Garcia, A. Giret and V. Botti

  Software engineering for Service-oriented MAS

  Twelfth International Workshop on Cooperative Information Agents (CIA08) Lecture Notes pp. In Press. (2008)

  Indexed in Computer Science Conference Ranking (0.55)

- Emilia Garcia, A. Giret and V. Botti

  Towards an evaluation framework for MAS software engineering

  Pacific Rim International conference on Multi-Agent (PRIMA) Lecture Notes in Artificial Intelligence (LNAI) pp. In Press. (2008)

  Indexed in Core Conference Ranking (B)

- Emilia Garcia, A. Giret and V. Botti

  On the evaluation of MAS development tools

  IFIP International Federation for Information Processing Artificial Intelligence and Practice II Max Bramer; (Boston: Springer), pp. 35-44 (2008)

- Emilia Garcia, E. Argente and A. Giret

  Issues for Organizational Multiagent Systems Development

  Sixth International Workshop From Agent Theory to Agent Implementation (AT2AI-6) pp. 59-65. (2008)

- Emilia Garcia, A. Giret and V. Botti

  Evaluating MAS Engineering Tools

  International Conference on Evaluation of Novel Approaches to Software Engineering pp. 181-184. (2008)

  Indexed in Core Conference Ranking (A)

- E. Del Val and M. Emilia Garcia and A. Espinosa and A. García-Fornes

  Herramientas de Depuración en Sistemas Multiagente

  IWPAAMS 07 pp. 41-50 isbn = "978-84-611-8858-1 (2007)

  Indexed in Computer Science Conference Ranking (0.56)

- S. Valero, E. Garcia, E. Argente, A. Giret, and V. Julian.

  FAST: a Flexible and Adaptable Scheduling Tool based on MAS Technology.

  Proceedings of the International Joint Conference Iberamia/SBIA/SBRN, 2006. 24.

# Bibliografía

[Alberola et al., 2007] Alberola, J. M.; Mulet, L.; Such, J. M.; García-Fornes, A.; Espinosa, A.; and Botti, V. 2007. Operating system aware multiagent platform design. In *Proceedings of Fifth European Workshop On Multi-Agent Systems (EUMAS 2007)*, 658–667. Association Tunisienne D'Intelligence Artificielle.

[Arcos et al., 2005] Arcos, J.; Esteva, M.; Noriega, P.; Rodríguez, J. A.; and Sierra, C. 2005. Environment Engineering for Multi Agent Systems. *Journal on Engineering Applications of Artificial Intelligence* 18:191–204.

[Argente et al., 2004] Argente, E.; Giret, A.; Valero, S.; Julian, V.; and Botti, V. 2004. Survey of MAS Methods and Platforms focusing on organizational concepts. In Vitria, J., Radeva, p. and Aguilo, I., ed., *Recent Advances in Artificial Intelligence Research and Development*, Frontiers in Artificial Intelligence and Applications, 309–316.

[Argente et al., 2007] Argente, E.; Palanca, J.; Aranda, G.; Julian, V.; Botti, V.; García-Fornes, A.; and Espinosa, A. 2007. *Supporting Agent Organizations*. 236–245.

[Argente, Julian, and Botti, 2006] Argente, E.; Julian, V.; and Botti, V. 2006. Multi-agent system development based on organizations. *Electronic Notes in Theoretical Computer Science* 150:55–71.

[Argente, Julian, and Botti, 2008] Argente, E.; Julian, V.; and Botti, V. 2008. Mas modelling based on organizations. In *9th Int. Workshop on Agent Oriented Software Engineering (AOSE08)*, 1–12.

[Argente, 2008] Argente, E. 2008. *GORMAS: Guías para el desarrollo de sistemas multiagente abiertos basados en organizaciones.* Ph.D. Dissertation, Departamento de Sistemas Informaticos y Computacion, Universidad Politecnica de Valencia.

[Bade et al., 2008] Bade, D.; Braubach, L.; Pokahr, A.; and Lamersdorf, W. 2008. An awareness model for agents in heterogeneous environments. In *Sixth International Workshop on Programming Multi-Agent Systems (ProMAS'08).*

[Bajo et al., 2007] Bajo, J.; Julian, V.; Corchado, J.; Carrascosa, C.; de Paz, Y.; Botti, V.; and de Paz, J. 2007. *Integrating new behaviours into the ARTIS agent architecture*, volume 1. Juan Manuel Corchado. 215–242.

[Bellifemine, Caire, and Greenwood, 2007] Bellifemine, F. L.; Caire, G.; and Greenwood, D. 2007. *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology).* John Wiley and Sons.

[Bernon, Cossentino, and Pavón, 2005] Bernon, C.; Cossentino, M.; and Pavón, J. 2005. An overview of current trends in european aose research. *Informatica (Slovenia)* 29(4):379–390.

[Bitting, Carter, and Ghorbani, 2003] Bitting, E.; Carter, J.; and Ghorbani, A. A. 2003. Multiagent System Development Kits: An Evaluation. In *Proc. of the 1st Annual Conference on Communication Networks and Services Research (CNSR 2003)*, volume May 15-16, 80–92.

[Boissier et al., 2006] Boissier, O.; Padget, J.; Dignum, V.; Lindemann, G.; Matson, E.; Ossowski, S.; Sichman, J.; and Vazquez-Salceda,

J. 2006. *Coordination, Organizations, Institutions and Norms in Multi-Agent Systems*, volume 3913 of *LNCS (LNAI)*.

[Boissier, Hübner, and Sichman, 2007] Boissier, O.; Hübner, J. F.; and Sichman, J. S. 2007. Organization oriented programming: From closed to open organizations. *Engineering Societies in the Agents World VII* 4457/2007:86–105.

[Bordini, Dastani, and Winikoff, 2007] Bordini, R. H.; Dastani, M.; and Winikoff, M. 2007. Current issues in multi-agent systems development (invited paper). *Proc. Workshop on Engineering Societies in the Agents World.* 38–61.

[Botti and Giret, 2008] Botti, V., and Giret, A. 2008. Anemona. a multi-agent methodology for holonic manufacturing systems. *Springer Series in Advanced Manufacturing* XVI:214.

[Braubach, Pokahr, and Lamersdorf, 2006] Braubach, L.; Pokahr, A.; and Lamersdorf, W. 2006. Tools and standards. *Multiagent Systems. Intelligent Applications and Flexible Solutions* 503–530.

[Braubach, Pokahr, and Lamersdorf, 2008] Braubach, L.; Pokahr, A.; and Lamersdorf, W. 2008. A universal criteria catalog for evaluation of heterogeneous agent development artifacts. In *Sixth International Workshop From Agent Theory to Agent Implementation (AT2AI-6)*.

[Bresciani et al., 2004] Bresciani, P.; Perini, A.; Giorgini, P.; Giunchiglia, F.; and Mylopoulos, J. 2004. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* 8(3):203–236.

[Carrascosa et al., 2003] Carrascosa, C.; Rebollo, M.; Soler, J.; Julian, V.; and Botti, V. 2003. Simba architecture for social real-time domains.

In *EUMAS 2003: The First European Workshop on Multi-Agent Systems*, 0–0. Agent Link.

[Cernuzzi and Rossi, 2002] Cernuzzi, L., and Rossi, G. 2002. On the evaluation of agent oriented modeling methods. In *In Proceedings of Agent Oriented Methodology Workshop*.

[Coutinho, Sichman, and Boissier, 2005] Coutinho, L.; Sichman, J.; and Boissier, O. 2005. Modeling Organization in MAS: A Comparison of Models. In *First Workshop on Software Engineering for Agent-oriented Systems*, 1–10.

[Criado et al., 2007] Criado, N.; Argente, E.; Julian, V.; and Botti, V. 2007. Organizational services for spade agent platform. In *IWPAAMS07*.

[Dam, 2003] Dam, K. H. 2003. Evaluating and Comparing Agent-Oriented Software Engineering Methodologies. Master's thesis, Master of Applied Science in Information Technology - RMIT University, Australia.

[Dastani et al., 2004] Dastani, M.; Hulstijn, J.; Dignum, F.; and Meyer, J.-J. 2004. Issues in multiagent system development. *Proceedings AAMAS'04*.

[Dignum and Dignum, 2006] Dignum, V., and Dignum, F. 2006. A landscape of agent systems for the real world. Technical report 44-cs-2006-061, Institute of Information and Computing Sciences, Utrecht University.

[Dignum et al., 2007] Dignum, F.; Dignum, V.; Thangarajah, J.; Padgham, L.; and Winikoff, M. 2007. Open agent systems ? *Eighth International Workshop on Agent Oriented Software Engineering (AOSE) in AAMAS07*.

[ecl, 2008] 2008. Eclipse - an open development platform.

[Eiter and Mascardi, 2002] Eiter, T., and Mascardi, V. 2002. Comparing environments for developing software agents. *AI Commun.* 15(4):169–197.

[Escriva et al., 2006] Escriva, M.; Palanca, J.; Aranda, G.; García-Fornes, A.; Julian, V.; and Botti, V. 2006. A jabber-based multi-agent system platform. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, 1282–1284. Association for Computing Machinery, Inc. (ACM Press).

[Esteva et al., 2001] Esteva, M.; Rodriguez-Aguilar, J.; Sierra, C.; Arcos, J.; and Garcia, P. 2001. *On the Formal Specification of Electronic Institutions.* Lecture Notes in Artificial Intelligence 1991. Springer-Verlag. 126–147.

[Esteva et al., 2004] Esteva, M.; Rosell, B.; Rodriguez, J. A.; and Arcos, J. L. 2004. AMELI: An agent-based middleware for electronic institutions. In *In Proc. of AAMAS04*, 236–243.

[Ferber, Gutknecht, and Michel, 2004] Ferber, J.; Gutknecht, O.; and Michel, F. 2004. From Agents to Organizations: an Organizational View of Multi-Agent Systems. In Giorgini, P.; Muller, J.; and Odell, J., eds., *Agent-Oriented Software Engineering VI*, volume LNCS 2935 of *Lecture Notes in Computer Science*, 214–230. Springer-Verlag.

[Garcia et al., 2008] Garcia, E.; Valero, S.; Argente, E.; Giret, A.; and Julian, V. 2008. A FAST method to achieve Flexible Production Programming Systems. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews* 38(2):242–252.

[Giorgini et al., 2005] Giorgini, P.; Mylopoulos, J.; Perini, A.; and Susi, A. 2005. The tropos metamodel and its use. *Informatical journal.*

[Giorgini, Kolp, and Mylopoulos, 2006] Giorgini, P.; Kolp, M.; and Mylopoulos, J. 2006. Multi-agent architectures as organizational structures. *Autonomous Agents and Multi-Agent Systems* 13(1):3–25.

[Giret, Botti, and Valero, 2005] Giret, A.; Botti, V.; and Valero, S. 2005. *MAS Methodology for HMS*, volume LNAI 3593. Springer Verlag. 39–49.

[Giret, 2005] Giret, A. S. 2005. *ANEMONA: Una Metodologia Multiagente para Sistemas Holonicos de Fabricacion.* Ph.D. Dissertation, Departamento de Sistemas Informaticos y Computacion, Universidad Politecnica de Valencia.

[Gomez-Sanz and Pavon, 2005] Gomez-Sanz, J., and Pavon, J. 2005. Ingenias development kit (idk) manual, version 2.5.

[Grasia, 2005] Grasia. 2005. INGENIAS IDE. http://ingenias.sourceforge.net/.

[Greenwood et al., 2007] Greenwood, D.; Lyell, M.; Mallya, A.; and Suguri, H. 2007. The ieee fipa approach to integrating software agents and web services. *Proc. AAMAS Industrial Track.*

[Hao, W., and Zhang, 2005] Hao, Q.; W., S.; and Zhang, Z. 2005. An autonomous agent development environment for engineering applications. *Advanced Engineering Informatics* 19:123–134.

[Horling and Lesser, 2005] Horling, B., and Lesser, V. 2005. Using odml to model multi-agent organizations. In *IAT '05: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology.*

[Howden et al., 2001] Howden, N.; Rönnquist, R.; Hodgson, A.; and Lucas, A. 2001. JACK Intelligent Agents-Summary of an Agent Infrastructure. In *Proc.of the 5th ACM Int.Conf. on Autonomous Agents.*

[Hubner, Sichman, and Boissier, 2006] Hubner, J.; Sichman, J.; and Boissier, O. 2006. S-moise+: A middleware for developing organised multi-agent systems. In *In Proc.Int. Workshop on Organizations in Multi-Agent Systems, from Organizations to Organization Oriented Programming in MAS*, volume 3913 of *LNCS*, 64–78.

[Huhns et al., 2005] Huhns, M.; Singh, M.; Burstein, M.; Decker, K.; Durfee, E.; Finin, T.; Gasser, L.; Goradia, H.; Jennings, N. R.; Lakartaju, K.; Nakashima; H.; Parunak, V.; Rosenschein, J.; Ruvinsky, A.; Sukthankar, G.; Swarup, S.; Sycara, K.; Tambe, M.; Wagner, T.; and Zavala, L. 2005. Research directions for service-oriented multiagent systems. In *IEEE Internet Computing*, volume 9(6), 52–58.

[ISO, 1999] ISO. 1999. Information technology - software product evaluation part 6: Documentation of evaluation modules. *ISO 14598-1:1999*.

[ISO, 2001] ISO. 2001. Software engineering - product quality - part 1: Quality model. *ISO IEC 9126-1:2001 edition*.

[ISO, 2005] ISO. 2005. Software engineering – software product quality requirements and evaluation (square) – guide to square. *ISO 25000:2005 edition*.

[iso, 2008] 2008. Iso - international organization for standarization http://www.iso.org/.

[ISO/IEC, 2006] ISO/IEC. 2006. Ergonomics of human-systeminteraction-part 110: Dialogue principles. *ISO9241-110:2006 edition*.

[Jack agent platform, 2008] Jack agent platform. 2008. http://www.agent-software.com/shared/products/index.html.

[Julian and Botti, 2003] Julian, V., and Botti, V. 2003. Estudio de metodos

de desarrollo de sistemas multiagente. *Revista Iberoamericana de Inteligencia Artificial* 81–96.

[Julian and Botti, 2004] Julian, V., and Botti, V. 2004. Developing real-time multiagent systems. *Integrated Computer-Aided Engineering* 11:135–149.

[Julian et al., 2000] Julian, V.; González, M.; Rebollo, M.; Carrascosa, C.; and Botti, V. 2000. Inside una herramienta para el desarrollo de agentes artis. In *Actas de SEID 2000*, volume I, 79–87. Senén Barro, José María Busta, Juan Manuel Corchado, Pedro Cuesta (eds.).

[Julian, 2002] Julian, V. J. 2002. *RT-MESSAGE: Desarrollo de Sistemas Multiagente de Tiempo Real*. Ph.D. Dissertation, Universidad Politecnica de Valencia. Departamento de Sistemas Informaticos y Computacion.

[Kostic, 2006] Kostic, M. 2006. *Code generation from AML Implementation into CASE tools and support for existing agent platforms*. Ph.D. Dissertation.

[Lin et al., 2007] Lin, C.-E.; Kavi, K. M.; Sheldon, F. T.; Daley, K. M.; and Abercrombie, R. K. 2007. A methodology to evaluate agent oriented software engineering techniques. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, 60. IEEE Computer Society.

[Lopez et al., 2007a] Lopez, J. S.; Bustos, F. A.; Rebollo, M.; and Julian, V. 2007a. Multiagent recommender system: A collaborative social network model. In *IWPAAMS07*, 159–168.

[Lopez et al., 2007b] Lopez, J. S.; Bustos, F. A.; Rebollo, M.; and Julian, V. 2007b. Multiagent recommender system: A tourism approach. In *INADIS 2007*, 51–61.

[Lopez, Bustos, and Julian, 2007] Lopez, J. S.; Bustos, F. A.; and Julian,

V. 2007. Tourism services using agent technology: A multiagent approach. *INFOCOMP - Journal of Computer Science - Special Edition* 51–57.

[Lopez, Luck, and d'Inverno, 2006] Lopez, F.; Luck, M.; and d'Inverno, M. 2006. A normative framework for agent-based systems. *Computational and Mathematical Organization Theory* 12:227–250.

[Mao and Yu, 2005] Mao, X., and Yu, E. 2005. Organizational and social concepts in agent oriented software engineering. In *AOSE IV*, volume 3382 of *Lecture Notes in Artificial Intelligence*, 184–202.

[Nwana et al., 1999] Nwana, H. S.; Ndumu, D. T.; Lee, L. C.; and Collis, J. C. 1999. Zeus: a toolkit and approach for building distributed multi-agent systems. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, 360–361. New York, NY, USA: ACM Press.

[OMG, 2002] OMG, O. M. G. 2002. Software Process Engineering Metamodel Specification Version 1.0. *http://www.omg.org/docs/formal/02-11-14.pdf*.

[Ossowski et al., 2007] Ossowski, S.; Julian, V.; Bajo, J.; Billhardt, H.; Botti, V.; and Corchado, J. 2007. Open issues in open mas: An abstract architecture proposal. In *CAEPIA 2007*, volume II, 151–160. AEPIA.

[P et al., 2000] P, B.; T, M.; S, R.; A, T.; S, T.; and S, P. 2000. Scalability in multi-agent systems: the fipa-os perspective. *Foundations and Applications of Multi-Agent Systems. UKMAS Workshop* 2403:110–130.

[Padgham and Winikoff, 2004a] Padgham, L., and Winikoff, M. 2004a. *Developing intelligent agent systems: A practical guide.* John Wiley and Sons.

[Padgham and Winikoff, 2004b] Padgham, L., and Winikoff, M. 2004b. Prometheus: A Methodology for Developing Intelligent Agents. In Heidelberg, S. B. ., ed., *Agent-Oriented Software Engineering III: Third International Workshop, AOSE 2002, Bologna, Italy, July 15, 2002. Revised Papers and Invited Contributions*, Lecture Notes in Computer Science. Springer.

[Papazoglou and van den Heuvel, 2006] Papazoglou, M. P., and van den Heuvel, W. 2006. Business process development lifecycle methodology. *to appear in Communications of ACM.*

[Papazoglou et al., 2006] Papazoglou, M. P.; Traverso, P.; Dustdar, S.; Leymann, F.; and Krämer, B. J. 2006. 05462 service-oriented computing: A research roadmap. In *Service Oriented Computing (SOC).*

[Pavon and Gomez, 2003] Pavon, J., and Gomez, J. 2003. Agent Oriented Software Engineering with INGENIAS. *3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003) : V. Marik, J. Müller, M. Pechoucek:Multi-Agent Systems and Applications II, LNAI 2691* 394–403.

[Pavon, Gomez-Sanz, and Fuentes, 2005] Pavon, J.; Gomez-Sanz, J.; and Fuentes, R. 2005. *The INGENIAS Methodology and Tools*, volume chapter IX. Henderson-Sellers. 236–276.

[Penserini et al., 2006] Penserini, L.; Perini, A.; Susi, A.; and Mylopoulos, J. 2006. From stakeholder needs to service requirements specifications. Technical report, itc-irst, Automated Reasoning Systems.

[P.Singh and N.Huhns, 2005] P.Singh, M., and N.Huhns, M. 2005. *Service-Oriented Computing Semantics, Processes, Agents.* John Wisley and Sons Ltd.

[Punter et al., 2004] Punter, T.; Kusters, R.; Trienekens, J.; Bemelmans, T.; and Brombacher, A. 2004. The w-process for software product

evaluation: A method for goal-oriented implementation of the iso 14598 standard. *Software Quality Control* 12(2):137–158.

[Rolland, Souveyet, and Kraeim, 2008] Rolland, C.; Souveyet, C.; and Kraeim, N. 2008. An intentional view of service-oriented computing. *Revue Ingénierie des Systèmes dÍnformation (ISI),RSTI (Revue des Sciences et Technologies de lÍnformation)- ISI - 13* 1:107 – 137.

[Shehory and Sturm, 2001] Shehory, O., and Sturm, A. 2001. Evaluation of modeling techniques for agent-based systems. *Agents-01* 624–631.

[Sierra et al., 2007] Sierra, C.; Thangarajah, J.; Padgham, L.; and Winikoff, M. 2007. Designing institutional multi-agent systems. *Agent-Oriented Software Engineering VII* 4405/2007:84–103.

[Software, 2004] Software, A. O. 2004. Jack intelligent agents: Jack teams manual, release 4.1.

[sta, 2008] 2008. Staruml the open source uml/mda platform.

[Sturm and Shehory, 2004] Sturm, A., and Shehory, O. 2004. A framework for evaluating agent-oriented methodologies. In *AOIS*, volume 3030 of *LNCS*, 94–109. Springer.

[Such et al., 2007] Such, J. M.; Alberola, J. M.; Mulet, L.; García-Fornes, A.; Espinosa, A.; and Botti, V. 2007. Hacia el diseño de plataformas multiagente cercanas al sistema operativo. In *International Workshop on Practical Applications of Agents and Multiagent Systems*, 1–10.

[Sudeikat et al., 2005] Sudeikat, J.; Braubach, L.; Pokahr, A.; and Lamersdorf, W. 2005. Evaluation of agent-oriented software methodologies examination of the gap between modeling and platform. (revised selected papers). *AOSE-2004 at AAMAS04*.

[Tinnemeier, Dastani, and Meyer, 2008] Tinnemeier, N.; Dastani, M.; and Meyer, J.-J. 2008. Orwell's nightmare for agents? programming multi-agent organisations. In *Sixth International Workshop on Programming Multi-Agent Systems (ProMAS'08)*.

[Tran and Low, 2005] Tran, Q.-N., and Low, G. 2005. Comparison of ten agent-oriented methodologies. 341–367.

[Trencansky and Cervenka, 2005] Trencansky, I., and Cervenka, R. 2005. Agent modelling language (AML): A comprehensive approach to modelling mas. In *Informatica*, volume 29(4), 391–400.

[Tsai et al., 2007] Tsai, W.-T.; Wei, X.; Paul, R.; Chung, J.-Y.; Huang, Q.; and Chen, Y. 2007. Service-oriented system engineering (SOSE) and its applications to embedded system development. In *Agent-Oriented Software Engineering III: Third International Workshop, AOSE 2002, Bologna, Italy, July 15, 2002. Revised Papers and Invited Contributions*.

[Tsai, 2005] Tsai, W. T. 2005. Service-oriented system engineering: A new paradigm. In *SOSE '05: Proceedings of the IEEE International Workshop*, 3–8. Washington, DC, USA: IEEE Computer Society.

[UML, 2008] UML, A. 2008. Agent uml http://www.auml.org.

[Val et al., 2007] Val, E. D.; Garcia, M. E.; Espinosa, A.; and García-Fornes, A. 2007. Herramientas de Depuración en Sistemas Multiagente. *IWPAAMS 07* 41–50.

[Valero et al., 2006] Valero, S.; Garcia, E.; Argente, E.; Giret, A.; and Julian, V. 2006. FAST: a Flexible and Adaptable Scheduling Tool based on MAS Technology. *Proceedings of the International Joint Conference Iberamia/SBIA/SBRN*.

[Valero et al., 2007] Valero, S.; Garcia, E.; Argente, E.; Giret, A.; and Julian, V. 2007. Flexible and Dynamic Production Programming

Tool Based on MAS Technology. *INFOCOMP: Journal of Computer Science* 1–8.

[Witwicki and Durfee, 2008] Witwicki, S. J., and Durfee, E. H. 2008. Commitment-based service coordination. In *Proc. SOCASE*, Springer LNCS 5006.

[Wooldridge and Ciancarini, 2001] Wooldridge, M., and Ciancarini, P. 2001. Agent-Oriented Software Engineering: The State of the Art. In *AOSE01*, volume 1957/2001 of *LNCS*, 55–82. Springer.

[Wooldridge, Jennings, and Kinny, 2000] Wooldridge, M.; Jennings, N. R.; and Kinny, D. 2000. The Gaia Methodology for Agent-Oriented Analisys and Design. *Journal of Autonomous Agents and Multi-Agent Systems* 15.

[Wooldridge, 1997] Wooldridge, M. 1997. Agent-Based Software Enginnering. In *IEE Proceedings of Software Engineering*, volume 14, 26–37.

[Xue, Zeng, and Liding, 2006] Xue, X.; Zeng, J.; and Liding, L. 2006. Towards an engineering change in agent oriented software engineering. In *ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control*, 225–228. IEEE Computer Society.