

# Ingeniería de Sistemas Multiagente vía Instituciones Electrónicas

Carles Sierra, Juan A. Rodríguez-Aguilar  
Pablo Noriega, Josep Ll. Arcos  
Artificial Intelligence Research Institute, IIIA  
Spanish Council for Scientific Research, CSIC  
08193 Bellaterra, Barcelona, Spain.  
{sierra,jar,pablo,arcos}@iiia.csic.es

Marc Esteva  
Graduate School of Library and Information Science  
University of Illinois at Urbana-Champaign  
501 E. Daniel Street, Champaign, IL 61820  
esteva@uiuc.edu

## Resumen

A medida que la complejidad de las aplicaciones comerciales aumenta, en particular desde la aparición de Internet, existe una necesidad de incorporar abstracciones organizacionales que faciliten su diseño, desarrollo y mantenimiento. Las instituciones electrónicas están en el trasfondo de esta aproximación. Éstas proporcionan un análogo computacional de las organizaciones humanas en las que agentes, tanto software como humanos, que adoptan roles diversos interactúan para conseguir objetivos individuales y organizacionales. En este artículo presentamos un entorno integrado de desarrollo destinado a la ingeniería de una clase de sistemas distribuidos, los sistemas multiagente, como instituciones electrónicas.

**Palabras clave:** sistemas multiagente, instituciones electrónicas, ingeniería del software, subastas

# 1 Introducción

A medida que la complejidad de las aplicaciones comerciales aumenta, en particular desde la aparición de Internet, existe una necesidad de incorporar abstracciones organizacionales que faciliten su diseño, desarrollo y mantenimiento. Las instituciones electrónicas están en el trasfondo de esta aproximación. Éstas proporcionan un análogo computacional de las organizaciones humanas en las que agentes, tanto *software* [8] como humanos, que adoptan roles diversos interactúan para conseguir objetivos individuales y organizacionales. En este escenario, la tecnología de agentes es candidata a ayudar a la reducción de costes operacionales y a la aceleración del ciclo de desarrollo de producto al permitir a las compañías, representadas por agentes, un funcionamiento más ágil y coordinado. Las instituciones electrónicas se ofrecen como un nexo apropiado entre socios de negocio con intereses propios, coordinando, regulando y auditando sus colaboraciones.

Pero, ¿por qué instituciones electrónicas? La investigación y el desarrollo en sistemas basados en agentes se ha caracterizado por considerar agentes benevolentes inmersos en infraestructuras sin fallos y operando en dominios relativamente simples. Tales asunciones no resultan válidas al considerar *sistemas abiertos* [6] cuyos componentes son desconocidos de antemano, pueden cambiar y evolucionar en el tiempo, pudiendo ser tanto humanos con intereses individuales como agentes *software* desarrollados por diferentes partes. Así, los sistemas multiagente abiertos [8] (SMA) pueden ser considerados como sistemas distribuidos en los que (posiblemente) vastas y variables poblaciones de agentes, exhibiendo comportamientos diferentes (posiblemente maliciosos o fraudulentos), interactúan. Notemos que a diferencia de los sistemas distribuidos, la cooperación entre estos agentes no se fija en tiempo de diseño, sino que emerge en tiempo de ejecución.

El carácter *abierto*, junto con los intereses individuales, pueden conducir a comportamientos globales inesperados y caóticos de un SMA si no se introducen mecanismos de control. El reto consistirá en evitar la emergencia de dinámicas negativas en tiempo de ejecución, particularmente en aplicaciones críticas (p.e. mercados electrónicos, gestión colaborativa de proyectos, organizaciones virtuales, gestión de redes de aprovisionamiento). De aquí que el diseño y desarrollo de SMA abiertos se presente como una tarea altamente compleja que motiva la necesidad de introducir estructuras reguladoras que establezcan lo que los agentes pueden y no pueden hacer. En este

sentido, observamos que las sociedades humanas han tratado con éxito cuestiones de regulación vía el desarrollo de instituciones. Es por ello que defendemos la introducción de su versión electrónica, las *instituciones electrónicas* (IEs) [9], para dar forma al entorno donde los agentes interactúan mediante la introducción de restricciones artificiales que articulan tales interacciones. Nuestra experiencia en el desarrollo de SMA como IEs [12, 2] nos permite defender la validez de esta aproximación. Notaremos, sin embargo, que como se subraya en [4, 11], consideramos que los ingenieros requieren disponer de herramientas. El propósito de este artículo es presentar un entorno de desarrollo integrado para IEs que facilita a los ingenieros el diseño y desarrollo de SMA. Pero para ello, en primer lugar es necesario caracterizar con rigor nuestra noción de IE.

## 2 Instituciones Electrónicas

Tal como se establece en [10], las interacciones humanas están reguladas por *instituciones*, que representan las reglas del juego en una sociedad, incluyendo cualquier restricción (formal o informal) que los seres humanos diseñan para articular sus interacciones. Así, las instituciones constituyen el marco en el que las interacciones humanas tienen lugar, definiendo lo que los individuos pueden y no pueden hacer y bajo qué condiciones. Las organizaciones humanas y los individuos cumplen las reglas de las instituciones a fin de legitimar sus acciones y recibir soporte legal. El establecimiento de una estructura estable para las interacciones humanas se presenta como la razón de ser de las instituciones.

En este artículo defendemos la adopción de una estrategia mimética con el objeto de afrontar la complejidad en la ingeniería de SMA abiertos. En tal caso, si mantenemos que los SMA abiertos pueden ser efectivamente diseñados e implementados como IEs, ¿qué es una IE? A continuación identificamos las nociones centrales en las que fundamentamos nuestra concepción actual de institución electrónica:

- *Agentes y Roles*. Los agentes son los "jugadores" en una IE, interactuando por medio del intercambio de *actos de habla*<sup>1</sup> o *ilocuciones*, mientras que los roles se definen como patrones estandarizados de comportamiento. Todo agente en una IE debe adoptar algún(os) rol(es). La noción de rol está siendo

---

<sup>1</sup>Según la propuesta de Searle[13], quien postula que los actos de habla no son simplemente proposiciones que puede ser ciertas o falsas, sino intentos por parte del emisor de conseguir algún objetivo o intención.

progresivamente considerada por los investigadores en la comunidad de agentes [15]. Nuestra aproximación distingue entre roles institucionales y no institucionales (externos), además de entre agentes institucionales y no institucionales (externos). Mientras que los roles institucionales son aquellos que se establecen para mantener y garantizar las normas institucionales, los roles no institucionales son aquellos que deben cumplir con tales normas.

- *Marco dialógico.* Algunos aspectos de una institución tales como los objetos del mundo y el lenguaje empleado para la comunicación son fijos, constituyendo el contexto o marco de interacción entre agentes. Las IEs establecen los actos de habla aceptables definiendo la ontología<sup>2</sup> y el lenguaje común para la comunicación y la representación del conocimiento, los cuales son agrupados en lo que denominamos el marco dialógico. Al compartir un marco dialógico, permitimos que agentes heterogéneos intercambien conocimiento con otros agentes.
- *Escena.* Las interacciones entre agentes son articuladas a través de agrupaciones de agentes que denominamos *escenas*, regidas por un protocolo de comunicación. Consideramos el protocolo de una escena como una especificación de los posibles diálogos que los agentes pueden mantener. Notar, sin embargo, que el protocolo de comunicación definiendo las posibles interacciones en una escena está orientado a roles y no a agentes particulares. En otras palabras, una escena define un marco de interacción para agentes basado en roles.
- *Estructura performativa.* Las escenas se conectan componiendo un *workflow* denominado *estructura performativa*. La especificación de una estructura performativa describe cómo los agentes pueden pasar de una escena a otra, definiendo tanto las precondiciones para incorporarse a o abandonar una escena. La satisfacción de tales condiciones dependerá fundamentalmente de los roles que se permitan adoptar a cada agente y de sus *compromisos*<sup>3</sup> adquiridos a través de actos de habla ya realizados.

---

<sup>2</sup>Según la definición en [1]: *Una ontología para un cuerpo de conocimiento referido a una tarea o dominio particulares describe una taxonomía de conceptos para tal tarea o dominio que define la interpretación semántica del conocimiento.*

<sup>3</sup>Entenderemos por compromisos aquellas obligaciones de realizar alguna acción o de provocar un estado del mundo.

- *Reglas normativas.* Las acciones de los agentes en el contexto de una institución tienen consecuencias, típicamente en forma de compromisos que imponen obligaciones o restricciones en las acciones dialógicas de los agentes en las escenas donde éstos actúan o actuarán en el futuro. El propósito de las reglas normativas es condicionar el comportamiento de los agentes imponiendo obligaciones o prohibiciones. En este sentido, los agentes institucionales están obligados a llevar a cabo las acciones necesarias que aseguren que los agentes no institucionales cumplen las normas institucionales.

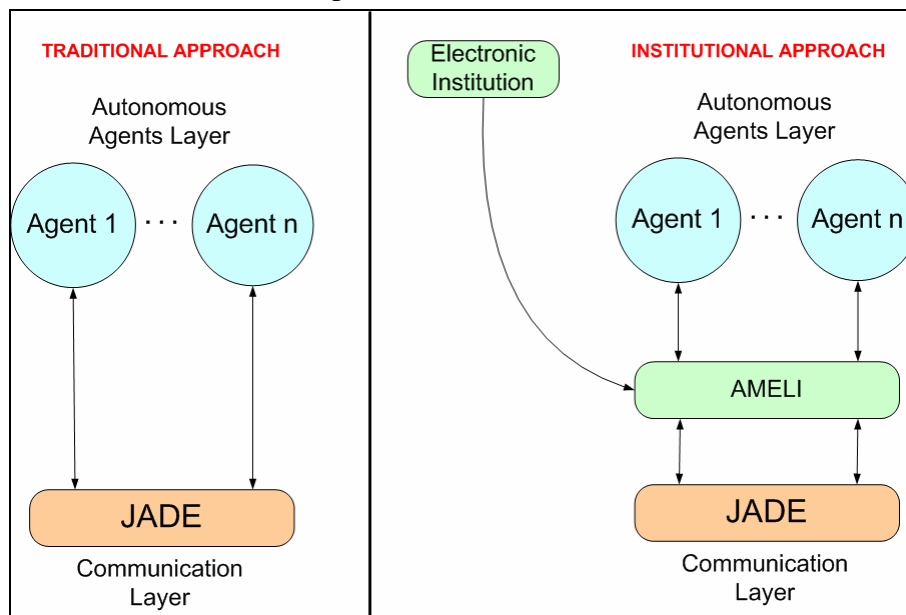


Figura 1: Mediación de agentes vía instituciones electrónicas

Los conceptos anteriores dibujan la estructura reguladora de una IE como un "workflow" (estructura peformativa) de protocolos multiagente (escenas) junto con una colección de reglas (normativas) que pueden ser activadas mediante las acciones de los agentes (actos de habla). En tiempo de ejecución, los agentes se incorporan y abandonan una IE, incorporándose a y abandonando escenas que son creadas y destruidas dinámicamente, y actuando mediante la realización de actos de habla. El objetivo principal de una IE es forzar el cumplimiento de las normas especificadas a los agentes participantes en tiempo de ejecución. A tal fin, todas las interacciones de los agentes son intermediadas por la IE como ilustra la figura 1. A diferencia del enfoque tradicional, que permite que los agentes interactúen abiertamente a través de una capa de comunicación, nuestra versión

computacional de una IE debe ser considerada como un *middleware* social entre los agentes externos participantes y la capa de comunicación escogida encargado de aceptar o rechazar sus acciones<sup>4</sup>.

### 3 Un Entorno de Desarrollo Integrado para Instituciones Electrónicas

IDE-eli, el entorno de desarrollo integrado para instituciones electrónicas, constituye un conjunto de herramientas destinadas a facilitar la ingeniería de SMA como instituciones electrónicas. Los agentes *software* aparecen como la tecnología de base de nuestra aproximación. Así, las instituciones electrónicas encapsulan los mecanismos de coordinación que intermedian las interacciones entre agentes *software* que representan a diferentes partes como se muestra en la figura 1. IDE-eli permite la ingeniería tanto de instituciones electrónicas como de sus agentes participantes. Destacaremos que desde el punto de vista de programación IDE-eli se aleja de un enfoque orientado a la máquina para ofrecer un enfoque inspirado en conceptos organizacionales que refleje mejor nuestra manera de entender las aplicaciones distribuidas como los SMA. IDE-eli permite una aproximación *top-down*: en primer lugar la organización y en segundo lugar los individuos. IDE-eli se compone de:

**ISLANDER** Una herramienta gráfica que permite la especificación de las reglas y protocolos en una institución electrónica.

**AMELI** Plataforma *software* que permite la ejecución de instituciones electrónicas especificadas con *ISLANDER*.

**aBUILDER** Herramienta de desarrollo de agentes.

**SIMDEI** Herramienta de simulación para animar, visualizar y analizar especificaciones creadas con *ISLANDER* antes de iniciar las fases de desarrollo e implantación.

La figura 2 muestra el rol de las herramientas de IDE-eli en el ciclo de desarrollo de una institución electrónica. A destacar que consideramos dicho ciclo como un proceso iterativo de refinamiento completamente soportado por las herramientas de IDE-eli. A continuación detallamos los pasos del ciclo de desarrollo junto con los roles desempeñados por las herramientas de IDE-eli.

---

<sup>4</sup>En la figura 1, la capa de comunicación escogida es Jade[7].

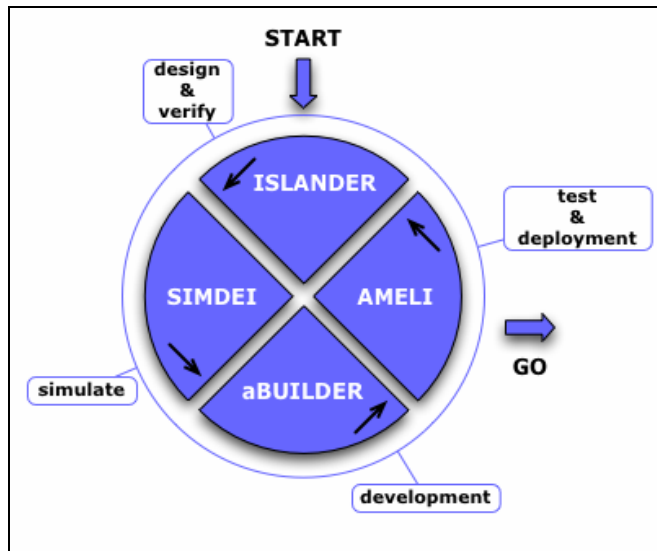


Figura 2: Ciclo de desarrollo de una institución electrónica.

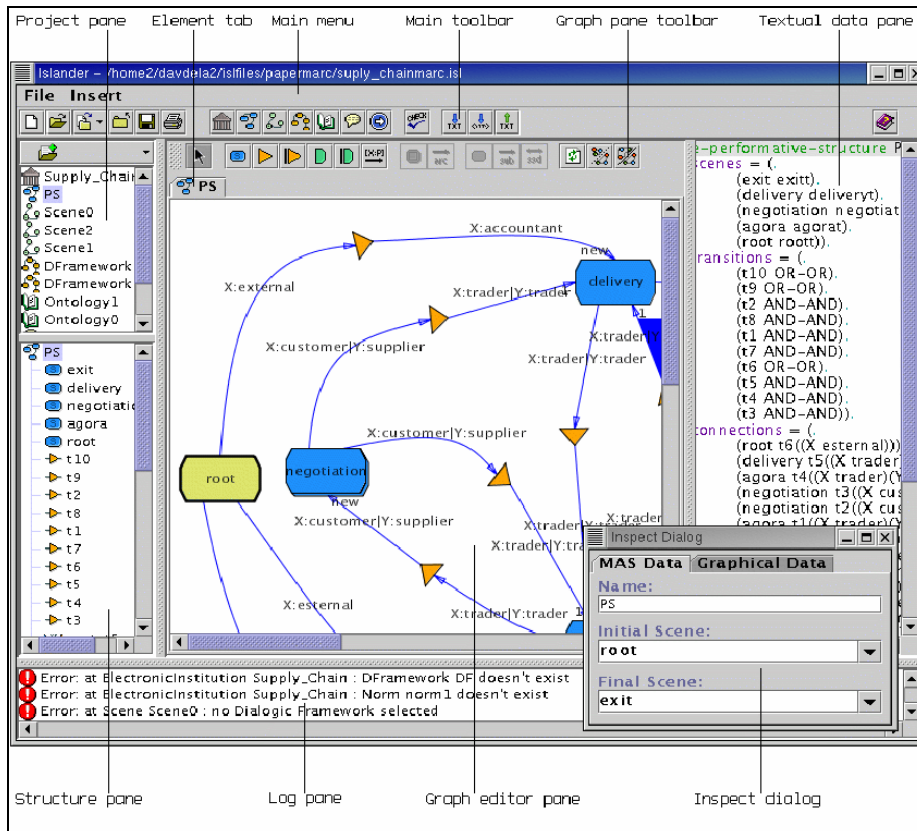


Figura 3: GUI de Islander.

- *Diseño.* Las instituciones electrónicas pueden ser gráficamente especificadas con la ayuda de *ISLANDER* [3]<sup>5</sup>. *ISLANDER* permite la definición de una ontología común, todas las interacciones que los agentes pueden llevar a cabo y las consecuencias de tales interacciones. El resultado es una descripción precisa de los tipos y el orden de los mensajes que los agentes en un SMA pueden intercambiar junto con una colección de normas que regulan tales interacciones. *ISLANDER* facilita la tarea de los diseñadores de IEs al permitir la especificación gráfica basada en la formalización de IEs realizada en [4]. Notamos que la especificación de una IE se centra en aspectos macro (sociales), en lugar de en aspectos micro (internos) de agentes. La figura 3 muestra la interfaz gráfica de *ISLANDER*.

- *Verificación.* Una vez especificada una institución, debería ser verificada antes de desarrollarla y activarla para permitir el acceso a agentes externos. Este paso se realiza en dos etapas. Mientras que la primera etapa analiza las propiedades estructurales, *estáticas*, la segunda etapa analiza el comportamiento esperado, *dinámico*, de la IE.

La *verificación estática* de IEs consiste en la comprobación de la validez estructural de las especificaciones. Por ejemplo, la comprobación de que los protocolos han sido correctamente especificados. Las funcionalidades necesarias para este proceso son ofrecidas por *ISLANDER*.

Por otro lado, la verificación dinámica de IEs se lleva a cabo vía simulación. El proceso se inicia con la definición de poblaciones de agentes de diferentes características capaces de operar en la IE especificada. A fin de facilitar esta tarea, hemos desarrollado un *generador de agentes* capaz de generar, a partir de una especificación, un esqueleto de agente dependiendo de los roles e interacciones en los que el agente pueda eventualmente participar. Para obtener un agente, los desarrolladores de agentes deben completar el esqueleto generado con mecanismos de decisión. Una vez desarrollados los agentes, es posible realizar simulaciones de la IE utilizando la herramienta de simulación *SIMDEI* desarrollada sobre *REPAST* [14]. *SIMDEI* permite simulaciones de IEs con poblaciones diversas de agentes con el

---

<sup>5</sup>Best prototype paper award at the First Joint International Conference on Autonomous Agents and Multiagent Systems AAMAS 2002.

objeto de realizar análisis *what-if*. El diseñador de la institución es responsable de analizar los resultados de las simulaciones y retornar a la etapa de diseño si éstos difieren de los esperados.

- *Desarrollo*. Una vez validada la especificación de una IE, ésta está lista para ser desarrollada y activada para la participación de agentes. Es el momento para los desarrolladores de agentes externos de implementar los agentes participantes. Es importante notar que no imponemos ninguna restricción en el tipo de agentes que pueden participar en una IE. Los desarrolladores de agentes pueden escoger sus propios lenguajes y arquitecturas. Sin embargo, consideramos que es importante facilitar este complejo proceso de desarrollo vía la herramienta *aBUILDER*. Ésta facilita el desarrollo de agentes en Java en una arquitectura predefinida. No obstante, en el futuro planeamos incorporar lenguajes y arquitecturas adicionales.
- *Implantación*. Una IE define un entorno normativo que articula las interacciones de agentes. Dado que los agentes pueden ser heterogéneos y tienen sus propios intereses, no podemos esperar que siempre se comporten de acuerdo a las reglas institucionales. Por lo tanto, toda IE es ejecutada vía *AMELI* [5], una infraestructura que intermedia y facilita las interacciones entre agentes al tiempo que obliga al cumplimiento de las reglas institucionales. La infraestructura implementada es de *propósito general* ya que puede *interpretar* cualquier especificación de *ISLANDER*. Por lo tanto, puede ser también considerada como *independiente del dominio*, pudiendo ser empleada en la implantación de cualquier IE especificada sin programación adicional. *AMELI* mantiene el estado de ejecución, utilizándolo junto con la especificación, para validar las acciones que los agentes intentan realizar. De aquí que la ejecución de una IE comience con la inicialización de *AMELI* una vez cargada la especificación. A partir de entonces, los agentes externos pueden acceder a la institución para interactuar con los otros agentes a través de *AMELI* (ver figura 1). La ejecución de una IE puede ser monitorizada gracias a una herramienta de monitorización que visualiza todos los eventos que ocurren en tiempo de ejecución tal como muestra la figura 4.

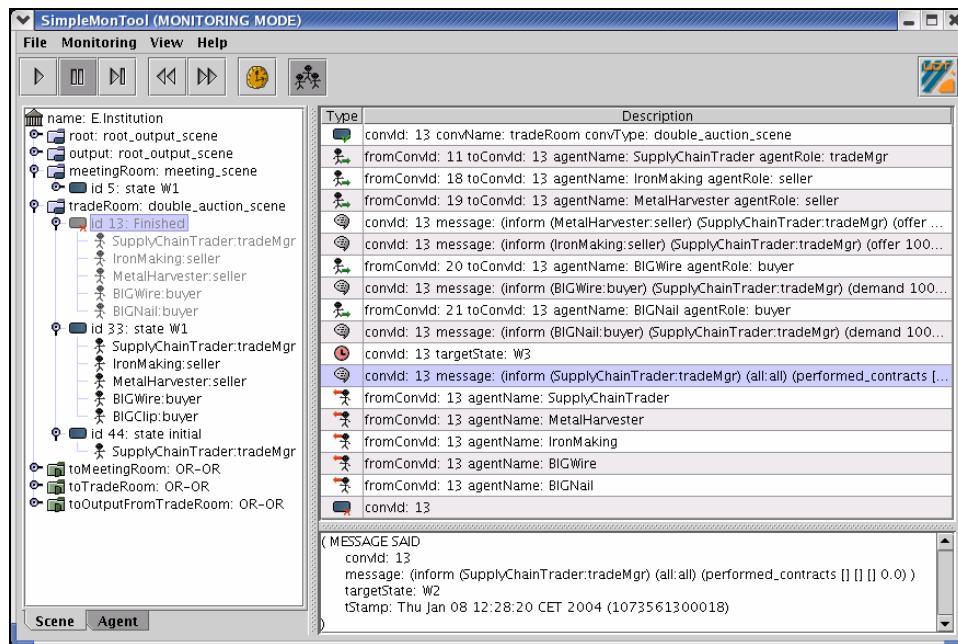


Figura 4: Monitorización de institución electrónica.

En la siguiente sección describimos de manera abreviada una IE real cuyo desarrollo ha sido realizado con la ayuda de IDE-eli.

## 4 Caso de Estudio: Agentes en Mercados Electrónicos

En la costa mediterránea tradicionalmente se ha vendido el pescado fresco mediante subastas a la baja realizadas en las lonjas junto a los puertos. Las cajas de pescado se agrupan en *lotes* que se subastan siguiendo el protocolo de la *subasta holandesa*: la lonja fija un precio de salida que va descendiendo rápidamente –cuatro precios por segundo– hasta que un comprador puja o el precio alcanza el valor de retirada. El comprador que puja primero siempre puede elegir entre quedarse todo el lote o solamente unas cajas, caso en el que se reinicia la subasta con la cantidad restante del lote. La subasta termina cuando todos los lotes se han vendido.

El mercado del pescado está adaptando sus métodos de venta a las nuevas tecnologías, y de hecho la mayor parte de las lonjas utilizan ya un sistema de subasta automatizado. Sin embargo, sigue siendo necesaria la presencia física de los compradores en las lonjas. Esto impone dos limitaciones claras. Por un lado, restringe los potenciales

compradores a aquellos que se encuentran presentes en la lonja. Por otro, supone que la participación simultánea en varias lonjas suponga un coste añadido de representantes en cada lonja. La eliminación de dichas limitaciones supondría grandes ventajas tanto para los compradores como para los vendedores. Al aumentar el número de compradores, el mercado se torna más competitivo y flexible, con los consiguientes ajustes en los precios o en el valor añadido de los productos. Además, se eliminarían los costes de los intermediarios.

La tecnología de agentes puede ser empleada para eliminar las limitaciones previamente mencionadas. El *Sistema Multi-Agente para el Comercio de Pescado* (MASFIT<sup>6</sup> en sus siglas inglesas) [2] permite a los compradores participar de forma remota y simultánea en varias lonjas por medio de agentes *software* a la vez que se mantienen las subastas originales. La participación de los agentes *software* en las subastas está mediada por una IE. La institución de MASFIT controla los accesos de los compradores a las lonjas, les mantiene informados como si estuvieran físicamente presentes en las lonjas y envía sus pujas durante las subastas. Para conseguirlo, el sistema automatizado de subastas de las lonjas se ha conectado también a la institución desarrollada. MASFIT interconecta múltiples lonjas, dando estructura a una federación de las mismas. Es importante recalcar que el sistema MASFIT garantiza igualdad de condiciones para los compradores presentes en la subasta y para los remotos, tanto en la información disponible como en la gestión de las pujas. De hecho, el sistema no modifica ni altera en absoluto el proceso habitual de las subastas en las lonjas, donde compradores humanos y *software* participan simultáneamente.

Tal y como se explica en [2], las herramientas de IDE-eli han resultado fundamentales en el diseño y desarrollo del sistema MASFIT. Por un lado la institución electrónica MASFIT se ha especificado usando *ISLANDER*. Por otro, las interacciones entre los agentes de la institución están gestionadas por *AMELI*. Esto facilita la participación y comunicación entre agentes dentro de la institución, a la vez que asegura el cumplimiento de las reglas institucionales fijadas en la especificación. Para ello, *AMELI* carga las especificaciones de las instituciones tal y como son generadas por *ISLANDER*. *AMELI* permite, entre otras cosas, la ejecución distribuida de los agentes. Esto resulta de vital importancia, dado que los agentes han de ejecutarse en diferentes máquinas.

---

<sup>6</sup>Multi-Agent System for FIsh Trading. <http://www.masfit.net>

Finalmente, recalcar que el escenario de mercado creado por MASFIT complica el proceso de decisión de compra, por realizarse éste coordinando varias subastas que tienen lugar de manera simultánea. Los agentes han de gestionar grandes cantidades de información –incluso información incierta– y su proceso de razonamiento ha de ser lo suficientemente rápido como para reaccionar ante los cambios. Por todo ello, el sistema MASFIT también está dotado de herramientas para crear, adaptar, gestionar y entrenar agentes compradores *software*.

## 5 Resumen

En este artículo hemos presentado un entorno de desarrollo integrado para la ingeniería de sistemas multiagente como instituciones electrónicas. Grandes ventajas se derivan del uso de las herramientas de IDE-eli. Por un lado, éstas reducen el ciclo de desarrollo. La ingeniería de una institución electrónica requiere un bajo coste de implementación, ya que sólo se han de programar los agentes que participan en ella. La flexibilidad inherente de *ISLANDER* en el diseño de los mecanismos de coordinación favorece un mantenimiento rápido y sencillo: una vez realizados los cambios en una nueva especificación, ésta está preparada para ser ejecutada por *AMELI*, así como los agentes. Por otro lado, las herramientas de simulación de IDE-eli soportan análisis *what-if* de los diseños de la institución electrónica antes de su desarrollo, facilitando la localización de comportamientos inesperados que pudieran poner en peligro aplicaciones críticas. Además, el desarrollo tanto de los agentes externos (no institucionales) como internos (institucionales) para las instituciones especificadas es soportado por *aBUILDER*.

IDE-eli ha demostrado su gran valor en el desarrollo de aplicaciones de comercio electrónico, tal y como se ha mostrado en el sistema MASFIT descrito en la sección 4. Sin embargo, existen otras muchas áreas a las que se puede hacer frente con la ayuda de estas herramientas. En términos generales, la aproximación basada en instituciones electrónicas resulta especialmente adecuada en dominios complejos donde están implicadas muchas entidades participantes y se requiere un alto grado de coordinación y colaboración. Por ello, aplicaciones en áreas como gestión automatizada de *workflows*, la monitorización y la gestión de la automatización de plantas industriales, o la gestión de redes de aprovisionamiento ofrecen un alto potencial para ser desarrolladas como instituciones electrónicas.

Para más información y descarga de software, el lector puede consultar <http://e-institutions.iii.csic.es>.

## Agradecimientos

Marc Esteva disfruta de una beca postdoctoral Fulbright/MECD FU2003-0569. Este artículo ha sido parcialmente financiado por el proyecto Web-i (2) (TIC-2003-08763-C02-01). Los autores querrían agradecer la inestimable ayuda de Teresa Solchaga en tareas de edición y la valiosa contribución de la Unitat de Desenvolupament Tecnològic del IIIA en el desarrollo de IDE-eli.

## Referencias

- [1] L. K. Alberts. *YMIR: An Ontology for Engineering Design*. PhD thesis, University of Twente, 1993.
- [2] G. Cuní, M. Esteva, P. Garcia, E. Puertas, C. Sierra, and T. Solchaga. Masfit: Multi-agent systems for fish trading. In *16th European Conference on Artificial Intelligence (ECAI 2004)*, Valencia, Spain, August 2004.
- [3] M. Esteva, D. de la Cruz, and C. Sierra. Islander: an electronic institutions editor. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002)*, pages 1045–1052, Bologna, Italy, July 15-19 2002.
- [4] M. Esteva. *Electronic Institutions: from specification to development*. IIIA PhD Monography. Vol. 19, 2003.
- [5] M. Esteva, J. A. Rodríguez-Aguilar, B. Rosell, and J. Ll. Arcos. Ameli: An agent-based middleware for electronic institutions. In *Third International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS'04)*, New York, USA, July 19-23 2004.
- [6] C. Hewitt. Offices are open systems. *ACM Transactions of Office Automation Systems*, 4(3):271–287, 1986.
- [7] Java Agent Development Framework. <http://sharon.cselt.it/projects/jade>.
- [8] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-agent Systems*, 1:275–306, 1998.
- [9] P. Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor*. IIIA Phd Monography. Vol. 8, 1997.

- [10] D. North. *Institutions, Institutional Change and Economics Performance*. Cambridge U. P., 1990.
- [11] J. A. Rodríguez-Aguilar. *On the Design and Construction of Agent-mediated Electronic Institutions*. IIIA Phd Monography. Vol. 14, 2001.
- [12] J. A. Rodríguez-Aguilar, P. Noriega, C. Sierra, and J. Padget. Fm96.5 a java-based electronic auction house. In *Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology(PAAM'97)*, pages 207–224, 1997.
- [13] J. R. Searle. *Speech acts*. Cambridge U.P., 1969.
- [14] REPAST URL. <http://repast.sourceforge.net>.
- [15] F. Zambonelli, N. Jennings, and M. Wooldridge. Developing multiagent systems: The gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3):317–370, 2003.