

# TOAST: Un Testbed para Evaluar Modelos de Confianza en Organizaciones Virtuales\*

Roberto Centeno<sup>1</sup>, Ramón Hermoso<sup>1</sup>, Holger Billhardt<sup>1</sup>, Sascha Ossowski<sup>1</sup>

<sup>1</sup> Grupo de Inteligencia Artificial, Universidad Rey Juan Carlos, c/ Tulipán s/n  
28933 Móstoles, España  
{roberto.centeno, ramon.hermoso, holger.billhardt, sascha.ossowski}@urjc.es

**Abstract.** Los mecanismos de confianza y reputación han comenzado a ser un aspecto muy importante en el campo de los sistemas multi-agente (SMA) debido a la autonomía y a la gran influencia del desarrollo de sistemas abiertos y heterogéneos. En particular, las organizaciones virtuales (OVs) son definidas como SMA con ciertas restricciones estructurales, como pueden ser: roles, interacciones, normas, etc. En este trabajo presentamos el desarrollo de un testbed que aúna estos conceptos, permitiendo la simulación de OVs, así como la experimentación y evaluación de distintos tipos de modelos de confianza y reputación.

**Keywords:** Sistemas multi-agente, organizaciones virtuales, mecanismos de confianza, reputación, simuladores.

## 1. Introducción

El concepto de organización se ha convertido en los últimos años en un aspecto muy importante en el campo de los Sistemas Multi-Agente (SMAs). Es bien conocido que las propiedades y estructuras organizativas proporcionan una gran ventaja a la hora de construir software, puesto que permiten afrontar diseños complejos usando abstracciones relativamente simples [10]. Estos conceptos organizativos a menudo dotan al sistema de cierta estructura y pueden, a su vez, perfilar el comportamiento de los agentes que pertenezcan al mismo. Los agentes que se unen a una organización generalmente se ven afectados por restricciones tales como la posibilidad de jugar determinados *roles* impuestos por la organización en diversas *interacciones*. Además, estas abstracciones pueden verse complementadas con otras de más alto nivel como son diversos conjuntos de *normas*, y sus consiguientes mecanismos que impidan a los agentes su correspondiente trasgresión [4]. Así pues, llamaremos Organización Virtual (OV) a aquellos SMAs que posean este tipo de estructuras organizativas y sus consecuentes restricciones [12].

---

\* Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia de España con el proyecto TIN2006-14630-C03-02 (y su programa de becas FPI), así como por la Consejería de Educación de la Comunidad de Madrid y la Universidad Rey Juan Carlos con el proyecto URJC-CM-2006-CET-0300.

En los últimos años, el desarrollo de mecanismos de confianza y reputación ha crecido considerablemente, debido especialmente al diseño de sistemas cada vez más abiertos y heterogéneos. El uso de esta clase de mecanismos es de gran ayuda para los agentes a la hora de seleccionar a otros con los que interactuar en determinadas situaciones y así conseguir los objetivos fijados. Los mecanismos de confianza suelen basarse en dos aspectos: la *experiencia pasada* [7] y las opiniones de otros agentes acerca de terceros (*reputación*) [13]. Si bien el primero proporciona información más fiable que el segundo, necesita de más tiempo para ser fundamentada, mientras que este último ofrece una información casi instantánea pero adolece de fiabilidad en muchos casos [5]. Así pues, modelos híbridos combinando estos dos tipos se imponen para los distintos tipos de escenarios donde se puedan aplicar.

Las OV's pueden verse muy beneficiadas del uso de mecanismos de confianza y reputación, puesto que, a pesar de las restricciones estructurales o las abstracciones normativas que impiden a los agentes realizar determinadas acciones, la propiedad de autonomía hace que los agentes aún puedan elegir *qué acciones* llevar a cabo de entre el elenco que tienen a su disposición y *con quién realizarlas* para conseguir los objetivos marcados. Los mecanismos de confianza se usan como un medio para tomar decisiones que afectan a dichos "problemas" de *libertad* dentro de la organización. Obviamente, este enfoque no tiene sentido en sistemas altamente monitorizados donde algún tipo de *autoridad* centralizada controle todas las posibles acciones de un agente, impidiéndole ejecutar ninguna que no siga un patrón ya estipulado; en cambio, la dirección que sigue la construcción de SMAs en los últimos tiempos, y en particular los SMAs organizativos, hace que los mecanismos de confianza resulten necesarios.

La herramienta que se presenta en este artículo (TOAST) surge como una necesidad de experimentar de manera simulada entornos organizativos (OV's) donde probar los modelos computacionales de confianza diseñados teóricamente. Si bien existen otros bancos de pruebas donde experimentar con mecanismos de confianza y reputación, entre los que destaca ART-Testbed [6], no se adaptaban a nuestras necesidades, en el sentido de que no simulan organizaciones, sino la evolución de agentes individuales en un medio compartido. Mientras que en ART-Testbed se pretende observar el comportamiento del agente de manera individual en un entorno meramente competitivo, nosotros pretendemos evaluar nuestros modelos computacionales desde un punto de vista social y cooperativo. El objetivo es la simulación de una organización a la que los agentes pueden conectarse y actuar usando diversos tipos de modelos de confianza distintos, para evaluar no sólo los resultados de dichos tipos de interacciones de forma individual, sino también para observar la evolución global de la organización como entidad de mayor nivel y con propósito definido.

En la Sección 2 se explica muy brevemente la investigación anterior que ha motivado la construcción de la herramienta TOAST. En la Sección 3, se detalla cómo está diseñado el testbed así como su funcionamiento. La Sección 4 expondrá brevemente qué ventajas proporciona el uso de TOAST como herramienta de simulación; por último, la Sección 5 contendrá las conclusiones y las líneas de trabajo futuras.

## 2. Modelos de Confianza Organizativos

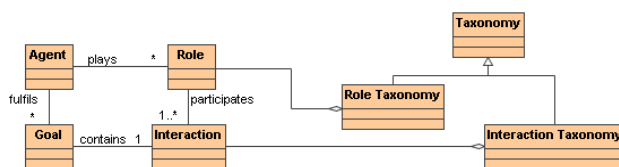
En trabajos anteriores [8,7], tratamos de realizar modelos teóricos de confianza basados en el desarrollo de agentes en marcos organizativos, más concretamente en OV. Estos trabajos parten de la clara necesidad de hacer más fiables y eficientes los modelos ya existentes. Éstos, en su mayoría basados en mecanismos de reputación, consiguen buenos resultados en escenarios determinados, pero son computacionalmente costosos, sobre todo, en entornos altamente escalables [5]. Así pues, nuestros modelos parten de la base de que la experiencia pasada del agente es muy valiosa, incluso para confiar o no en situaciones que no han sido evaluadas previamente. La mayoría de mecanismos, al evaluar la confianza para un agente determinado en una situación específica, deciden pedir opiniones a otros (con el riesgo que eso conlleva) si no tienen información previa de dicha situación para dicho agente. Nuestro enfoque va más allá. Parte de la hipótesis siguiente: "una vez miembros de una OV, los agentes tienden a comportarse de manera similar en situaciones similares" o, en términos de nuestra concepción de OV [8], los agentes tienden a comportarse de manera similar cuando juegan roles similares en interacciones similares. Así, con un enfoque probabilístico sencillo, y apoyándonos en la posibilidad de definir funciones de similitud a partir de las estructuras públicas de la organización (taxonomías de roles y de interacciones), hemos definido modelos de confianza que *inferen* en una gran cantidad de casos el comportamiento de otros agentes para situaciones en las que los agentes no tienen experiencia o la que tienen es escasa y poco fiable [7].

## 3. Trust Organisational Agent System Testbed (TOAST)

En esta sección vamos a presentar TOAST (*Trust Organisational Agent System Testbed*), la herramienta que hemos desarrollado para evaluar y comparar distintos modelos de confianza, en el marco de las organizaciones virtuales de agentes. TOAST está concebido para monitorizar la organización a nivel *micro* y a nivel *macro*. La confianza como concepto abstracto es difícil de medir, por lo que se suele tomar como *feedback* las consecuencias que tiene su uso a la hora de utilizarlo como medio de decisión. Así pues, midiendo la utilidad de las acciones derivadas del acto de *confiar*, podemos aproximar la bondad del mecanismo de confianza utilizado por el agente en cuestión (nivel *micro*). No obstante, el agente pertenece a una organización, por lo que la repercusión de sus decisiones puede afectar, asimismo, al resto (nivel *macro*). TOAST permitirá visualizar la evolución tanto individual como global de los agentes.

Con este fin, podemos simular organizaciones donde los *agentes* se unen, jugando una serie de *roles*, para realizar ciertas *interacciones* con otros miembros, con el fin de conseguir una serie de *objetivos* (ver Fig. 1). La organización por su parte, definirá *a priori* las interacciones que se pueden realizar dentro de ella, así como los roles permitidos para llevarlas a cabo. Esta información (roles e interacciones) será pública y disponible para todos los agentes que la requieran, y a su vez estará estructurada en

forma de *taxonomías* de conceptos para su posterior uso en los modelos de confianza que queremos evaluar [8].



**Fig. 1. Diagrama de una OV en TOAST**

Nótese que, por razones de simplicidad, hemos asumido que cada objetivo se asocia con una sola interacción para su consecución.

### 3.1 Definición de la organización virtual en TOAST

En primer lugar, tenemos que especificar el tipo de organización con el que vamos a trabajar, el dominio de la OV (*e.g.* organizaciones en el dominio universitario). Para ello, necesitamos definir los siguientes elementos:

- *Taxonomía de roles*: define las funcionalidades que pueden desempeñar los agentes en dicho tipo de organización.
- *Taxonomía de interacciones*: define las interacciones que se pueden realizar en ese tipo de organización, en términos de los roles necesarios para participar en ellas.
- *Tipos de objetivos*: definimos los tipos de objetivos que pueden cumplir los agentes que se unan a OV del dominio definido. Asociamos la interacción que deben realizar para conseguir cada tipo de objetivo.

Todos estos elementos se introducen en el simulador mediante ficheros XML.

### 3.2 Instanciación de la organización virtual en TOAST

Una vez tenemos definido el dominio del escenario que vamos a utilizar, el siguiente paso será la creación de una OV concreta del dominio definido anteriormente. En esta etapa, hay que definir el número y el(los) tipo(s) de agentes que se van a unir a ella, así como el tipo de modelo de confianza que van a utilizar y sus parámetros si los hubiera. Los tipos de agentes y los modelos de confianza son diseñados e implementados por el usuario usando las librerías e interfaces creadas a tal efecto. Así pues, los agentes serán simulados, no hacen uso de ninguna plataforma de programación de agentes, puesto que sería demasiado pesado para los experimentos básicos que queremos llevar a cabo. Se decidió, igualmente, no usar herramientas específicas de simulación como ([3]), puesto que su adaptación a nuestros requisitos era innecesaria.

Como hemos visto, los agentes se conectan a la organización con una serie de roles que pueden desempeñar potencialmente en ella. Una vez que tenemos los agentes que se van a conectar y los posibles roles que pueden desempeñar, podemos elegir los

porcentajes de dispersión de cada tipo de rol dentro de la organización (cuántos agentes lo van a poder desempeñar). Esta característica es de especial utilidad cuando lo que tratamos de evaluar son mecanismos de confianza, puesto que cuanto más distribuido esté un rol en la organización, mayor importancia tendrán dichos mecanismos a la hora de buscar compañeros de interacción o proveedores de servicio.

Otro elemento que debemos proporcionar al simulador para crear nuestra organización es el número de objetivos a generar para ser asignados a los distintos agentes. Estas secuencias generadas se pueden almacenar para su posterior uso, de manera que se puedan repetir los experimentos bajo las mismas condiciones.

A partir de estos elementos proporcionados, el simulador generará una serie de datos necesarios, de manera estocástica. Generará y asociará los roles que pueden desempeñar cada agente dentro de la organización. Estos roles se generarán teniendo en cuenta los porcentajes de propagación introducidos por el usuario. La asignación de los mismos se realizará de manera aleatoria entre los distintos agentes creados.

A partir de los datos de entrada, el simulador modelará el comportamiento que van a tener los agentes en la organización: la habilidad de cada agente jugando un rol concreto en cada tipo de interacción. Con este propósito, se generan los valores de habilidad (*capability values*). Para cada tupla  $\langle \text{Agente}, \text{Rol}, \text{Interacción} \rangle$  ( $\langle A, R, I \rangle$ ) en la que puede estar inmerso un agente, generamos una distribución normal, que modela el comportamiento del agente  $A$ , jugando el rol  $R$ , en la interacción  $I$ ,

$$Capability_{\langle A, R, I \rangle} \equiv N_{\langle A, R, I \rangle}(\mu, \sigma) \quad (1)$$

donde la media  $\mu$ , representa el comportamiento medio del agente en esa situación y la desviación típica  $\sigma$ , representa las variaciones que va a tener sobre ese comportamiento medio.

Como hemos comentado en la Sección 2, el enfoque de nuestros modelos de confianza está centrado en la visión local del agente, complementada con la inferencia que puede realizar éste a partir de las estructuras organizativas que conforman la organización. Así pues, para probar nuestras asunciones, los valores de habilidad para cada posible tupla  $\langle A, R, I \rangle$ , se generarán teniendo en cuenta nuestra hipótesis de partida: *los agentes se comportan de manera similar en situaciones similares* [7]. Para ello, se calcula en primer lugar, el valor de habilidad global para cada agente ( $Capability(A_i)$ ), definiendo el comportamiento general (para cualquier tipo de situación) de ese agente. A continuación, utilizando ese valor global del agente y teniendo en cuenta las similitudes de los roles (según la taxonomía), se genera, de manera correlada, un valor de habilidad para ese agente desempeñando los distintos roles que puede jugar ( $Capability(A_i, R_j)$ ). Por último, se generan los valores de habilidad para cada tupla ( $Capability(A_i, R_j, I_k)$ ), de manera correlada, a partir de los valores  $Capability(A_i, R_j)$ , y teniendo en cuenta la similitud de las interacciones definidas en la taxonomía.

Lo último que va a generar el simulador serán los distintos objetivos que tienen que lograr los agentes. Para ello, partiendo de la lista de tipos de objetivos que ha introducido el usuario y teniendo en cuenta los roles que tienen asignados los agentes, generará un conjunto de objetivos  $O = \{o_1, \dots, o_i, \dots, o_n\}$ , que asignará de manera aleatoria a cada agente.

### 3.3 Flujo de ejecución de TOAST

Para explicar el flujo de ejecución que sigue la herramienta, vamos a suponer un escenario donde simulamos una OV que implementa el funcionamiento de una universidad, con los roles típicos que aparecen en este tipo de organización (profesor, estudiante, etc.). Suponemos que está participando en la OV el agente A jugando el rol de *estudiante*:

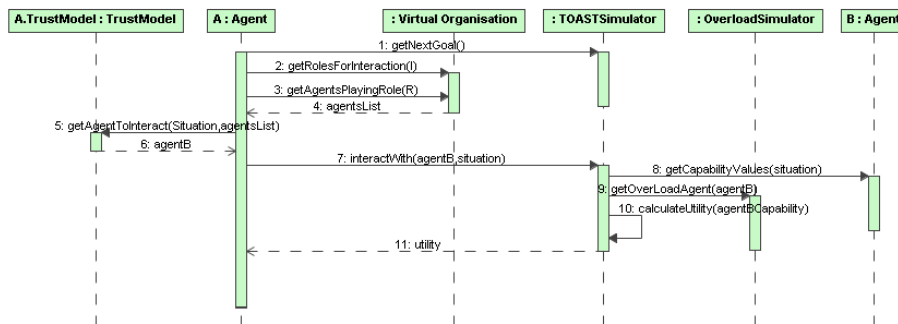


Fig. 2. Flujo de ejecución genérico de TOAST

1. En primer lugar, el agente A procesará el primer objetivo  $o_i \in O$  que tiene asignado en su cola de objetivos. Observará qué interacción necesita realizar para lograrlo, y por tanto qué rol debe desempeñar para realizar esa interacción. En nuestro ejemplo el primer objetivo que tendrá que lograr el agente será  $o_i \equiv$  *aclarar una duda sobre la asignatura IA*. Para conseguir este objetivo debe realizar la interacción  $I \equiv$  *llevar a cabo una tutoría de IA con un profesor*.
2. El agente A sabe que para llevar a cabo esa interacción los agentes deben desempeñar los roles de *profesor* y *estudiante*; esta información es pública para los miembros de la OV. En este caso, el agente A está jugando el rol  $R_1 \equiv$  *estudiante*, por lo que necesita encontrar un agente que esté jugando el rol  $R_2 \equiv$  *profesor*. Para ello, preguntará a la OV qué agentes están desempeñando dicho rol.
3. La organización le proporcionará una lista con los posibles candidatos (todos los agentes que pueden desempeñar el rol de *profesor*), entre los que podrá elegir el compañero de interacción.
4. A partir de esta lista de candidatos, utilizando su modelo de confianza, seleccionará un compañero final. El modelo intentará evaluar qué agente, de entre los candidatos, le ofrece más confianza para esa situación, es decir, qué agente le ofrece más confianza jugando el rol de *profesor* en la interacción *llevar a cabo una tutoría*. Para esto, podrá hacer uso de la reputación, es decir, podrá solicitar opinión a otros agentes sobre un tercero.
5. Una vez que el agente A ha elegido un compañero de interacción (supongamos que ha elegido al agente B). El siguiente paso será la simulación de la interacción.

TOAST simulará que el agente  $A$ , jugando el rol  $R_1$ , realizará la interacción  $I$ , con el agente  $B$  jugando el rol  $R_2$  ( $A \rightarrow \langle B, R_2, I \rangle$ ).

6. Como resultado de la simulación de la interacción, TOAST devolverá al agente  $A$  un valor de utilidad, representando el grado de satisfacción obtenido al realizar la interacción con el agente  $B$ . Este valor será generado siguiendo el siguiente proceso:

$$Utility_{A \rightarrow \langle B, R_2, I \rangle} = randomPM(Capability_{\langle B, R_2, I \rangle}) \quad (2)$$

donde  $Capability_{\langle B, R_2, I \rangle} \equiv N_{\langle B, R_2, I \rangle}(\mu^*, \sigma)$  y  $randomPM$ , es una función que genera un número aleatorio de una distribución normal utilizando el método *Polar de la Normal* [9]. Además, el comportamiento del agente podrá verse alterado por la sobrecarga que pueda sufrir, debida a una excesiva petición para participar en diversas interacciones. La nueva media  $\mu^*$ , que contiene ya esta posibilidad de sobrecarga, será calculada de la siguiente manera:

$$\mu^* = \mu - (\mu \cdot OL_B^t) \quad (3)$$

donde  $OL_B^t \in [0..1]$  representa la sobrecarga (*overload*) del agente  $B$  en el instante de tiempo  $t$ . Además de la sobrecarga, la media del comportamiento del agente podría verse también afectada por otra serie de parámetros inherentes a los propios modelos de confianza, como pueden ser la *disposición* hacia el otro agente, la *dependencia* que pueda tener de él, el *riesgo* que quiera asumir, etc.

7. Una vez que TOAST ha generado el valor de utilidad, con sus posibles modificaciones, se lo devuelve al agente  $A$  como resultado de su interacción con el agente  $B$ . Este valor podrá ser utilizado por el agente  $A$  como *feedback*, actualizando su modelo de confianza y sus estructuras internas.

Siguiendo este flujo de ejecución, van realizando paralelamente todos sus objetivos los distintos agentes de la OV. Simultáneamente, el simulador va almacenando todos los aspectos importantes (evolución de las utilidades, peticiones de reputación realizadas, errores en las estimaciones, etc.) para poder ser visualizados.

## 4. Experimentando con TOAST

La versión actual de TOAST nos permite comparar mecanismos de confianza basados en experiencias pasadas, basados en reputación, híbridos, así como basados en inferencia a partir de similitudes; todo esto mediante experimentos en el marco de las organizaciones virtuales de agentes<sup>1</sup>. Los experimentos se pueden parametrizar, *e.g.* añadiendo similitudes o disimilitudes entre los conceptos organizativos (roles e

---

<sup>1</sup> Aunque esta versión de TOAST está orientada a la experimentación con mecanismos de confianza, su estructura puede permitir evaluar otras características tanto organizativas como individuales de los propios agentes.

interacciones), simulando la sobrecarga de los agentes, etc. Podemos agrupar los datos de salida de TOAST según su naturaleza, como se explica a continuación:

**Evolución de la utilidad.** La utilidad representa el grado de satisfacción que obtiene cada agente en la interacción con otro miembro de la organización. Este nivel de satisfacción está relacionado con la calidad con la que realizan los agentes las diferentes interacciones. Dicho valor de utilidad, por ejemplo, podría ser equivalente a un valor económico obtenido por los agentes como resultado de las interacciones que realizan, permitiendo de esta manera que las utilidades sean comparables.

Como resultado de las ejecuciones del simulador, podemos observar la evolución de la utilidad conseguida, tanto a *nivel macro*, utilidad obtenida por la organización como entidad; como a *nivel micro*, utilidad conseguida por cada agente de forma independiente. La utilidad a nivel macro se puede computar mediante una función que se defina el propio usuario o programador, por ejemplo, mediante la agregación de las utilidades individuales de cada agente. No obstante, este grado de satisfacción organizativo no es más que una forma de normalizar el resultado global de los modelos para poder así ser comparados.

**Gestión de la reputación.** Otro aspecto importante para evaluar los modelos de confianza, es la gestión de las peticiones de opiniones (*mecanismo de reputación*) que realizan los agentes, con cada modelo. Una petición sobre la opinión que tiene un agente sobre un tercero siempre conlleva un riesgo. En los sistemas abiertos, donde los agentes que los forman pueden ser heterogéneos (especialmente en cuanto a su comportamiento), realizar peticiones de este tipo puede traer debilidades, debido a que pueden existir agentes maliciosos, agentes que mientan al dar sus opiniones, etc., con el consiguiente riesgo de pérdida de utilidad que pueden sufrir los agentes en estos casos. Igualmente, estas peticiones de reputación, pueden llevar asociado un coste, tanto computacional como de otra naturaleza.

Por tanto, aunque el uso de la reputación puede ayudarnos en nuestra tarea, se debe encontrar un equilibrio en el uso de ésta. Por este motivo, la herramienta permite observar, asimismo, cómo evolucionan a lo largo del tiempo las peticiones y respuestas acerca de reputación que se intercambian los agentes, con los distintos modelos de confianza.

**Datos de los agentes.** Para analizar más detalladamente cada modelo, TOAST nos permite observar detalles del estado de los agentes. Podemos ver los detalles *estáticos* de los mismos, como son los roles que pueden desempeñar, sus valores de habilidad, el valor de buena voluntad hacia los demás, cómo va a variar ese valor, etc. Por otro lado, podemos ver aspectos *dinámicos* de los agentes. Entre estos datos observamos: las transacciones que realiza cada agente (interacciones y solicitudes de reputación), el estado de sus estructuras de datos internas, cómo evolucionan las entradas de éstas, con qué agentes interactúa, cuántas veces con cada uno y en qué instante de tiempo, cuáles son sus fuentes de reputación (a qué agentes solicita opiniones de reputación), cuántas veces a cada uno y en qué instante, etc.

Con todos estos datos, podemos tener monitorizado a cada agente, permitiéndonos compararlos entre las distintas ejecuciones, es decir, qué datos obtiene un agente, en el mismo escenario, teniendo que realizar los mismos objetivos, ayudándose para ello de distintos modelos de confianza.



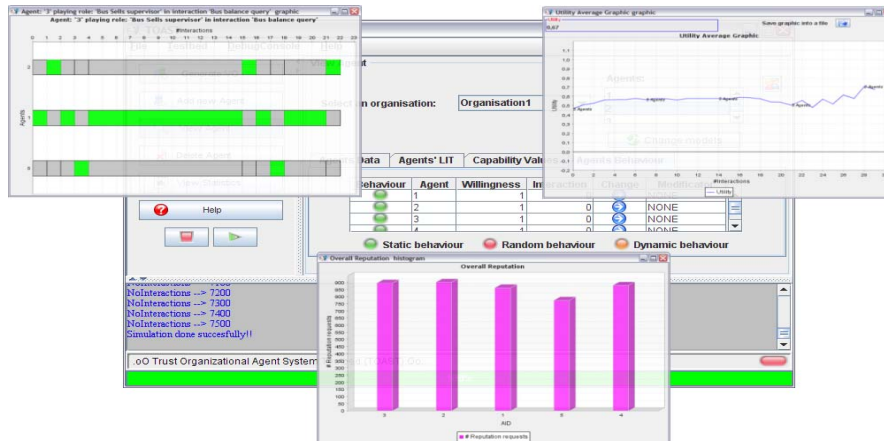


Fig. 3 Captura de la herramienta TOAST

## 5. Conclusiones

En este trabajo se describe TOAST, una herramienta para la simulación de organizaciones virtuales de agentes y la evaluación de mecanismos de confianza. El desarrollo de esta herramienta surge con la necesidad de evaluar mecanismos de confianza en entornos organizativos ya creados [8], donde no sólo importa la evolución individual de los agentes (como entidades interesadas “*self-interested*”) sino también cómo evoluciona globalmente la organización como entidad que desea cumplir sus propios objetivos. Para ello, TOAST nos permite la monitorización de las distintas simulaciones tanto a nivel *macro* (evolución de la organización) como a nivel *micro* (evolución de cada agente participante).

El entorno ha sido probado con distintos casos de estudio simples como son organizaciones de carácter académico, agencias de turismo y también en entornos orientados a servicios [1]. Si bien existen gran cantidad de modelos de confianza y reputación, también existen igual cantidad de *testbeds* para simularlos [11,14]. Cada uno de ellos está orientado a realizar experimentos enfocados al modelo específico que tratan de validar, por lo que la comparación de modelos se convierte en algo realmente complejo. Por otra parte, en los últimos años se han hecho esfuerzos por aunar un único *testbed* (ART Testbed [6]) que dé cabida a la experimentación con modelos heterogéneos de confianza y reputación. Sin embargo, ART está orientado a sociedades abiertas de agentes que no cuenten con una estructura organizativa compleja. En consecuencia, el entorno competitivo que desarrolla y sus limitaciones en cuanto al diseño (no existe el concepto de organización, ni tan siquiera de grupo) hacen que nuevos *testbeds* como TOAST sean necesarios.

Como líneas futuras ya se está trabajando en el diseño distribuido del simulador, de tal forma que acepte la experimentación con organizaciones escalables. Estas organizaciones permitirán la entrada y salida de agentes durante la simulación, así

como el cambio en tiempo real de los parámetros de los agentes, para evaluar cambios forzados de comportamiento.

## Referencias

1. H. Billhardt, R. Hermoso, S. Ossowski, and R. Centeno. Trust based service provider selection in open environments. In 22nd Annual ACM Symposium on Applied Computing (SAC2007), Seoul, Korea, March 11-15, pages 1375–1380, 2007.
2. C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In ICMAS '98: Proceedings of the 3rd International Conference on Multi Agent Systems, page 72, Washington, DC, USA, 1998. IEEE Computer Society.
3. N. Collier. RePast: an extensible framework for agent simulation, 2002.
4. M. Esteva, B. Rosell, J.A. Rodríguez-Aguilar, and J.LI. Arcos. AMELI: An agent-based middleware for electronic institutions. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, volume 1, pages 236–243, 2004.
5. K. Fullam and K.S. Barber. Dynamically learning sources of trust information: Experience vs. reputation. In Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'07), pages 1055 – 1062, 2007.
6. K. Fullam, T. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K.S. Barber, J.S. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In AAMAS'05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, pages 512–518, New York, NY, USA, 2005. ACM Press.
7. R. Hermoso, H. Billhardt, R. Centeno, and S. Ossowski. Effective use of organisational abstractions for confidence models. In O. Dikenelli, G. O'Hare, M. O'Grady and A. Ricci, editors, Engineering Societies in the Agents World, volume 4457 of LNAI, pages 368–383, Berlin, In press. Springer.
8. R. Hermoso, H. Billhardt, and S. Ossowski. Integrating trust in virtual organisations. In P. Noriega and J. Vázquez-Salceda, editors, Coordination, Organization, Institutions and Norms in Agent Systems II, volume 4386 of LNAI, pages 17–29, Berlin, In press. Springer.
9. D.E. Knuth. The Art of Computer Programming - Seminumerical Algorithms, volume 2. Reading, Massachusetts: Addison-Wesley, 3rd edition, 1998.
10. A. Omicini and S. Ossowski. Objective versus subjective coordination in the engineering of agent systems. In M. Klusch, S. Bergamaschi, P. Edwards, and P. Petta, editors, Intelligent Information Agents: An AgentLink Perspective, volume 2586 of LNAI: State-of-the-Art Survey, pages 179–202. Springer-Verlag, March 2003.
11. J. Sabater and C. Sierra. REGRET: a reputation model for gregarious societies. In Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada, pages 61-69, 2001.
12. M. Schumacher and S. Ossowski. The governing environment. In Weyns, Parunak, and Michel, editors, Environments for Multiagent Systems II, volume 3830, pages 88–104. Springer-Verlag, 2006.
13. W. T. Luke Teacy, J. Patel, N.R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.
14. G. Zacharia, A. Moukas and P. Maes. Collaborative Reputation Mechanisms in Electronic Marketplaces. In Proceedings of Hawaii International Conference On System Sciences (HICSS), 2002.