

Sistemas Multiagente de Tiempo Real

Iván Marsá Maestre, Sebastián Sánchez Prieto, and Juan R. Velasco

Departamento de Automática, Universidad de Alcalá, ESPAÑA
{ivmarsa,ssp,juanra}@aut.uah.es

Resumen Existen diferentes aplicaciones y entornos de tiempo real donde puede ser necesario o conveniente incluir capacidades de proceso e inferencia más propias de la inteligencia artificial. Este trabajo realiza un breve análisis de la problemática de integrar inteligencia artificial y consideraciones de tiempo real en una misma aplicación, así como de las técnicas más utilizadas para ello. Por último, se presentan los sistemas multiagente de tiempo real como una alternativa prometedora para este tipo de sistemas, se enumeran los requisitos que deben cumplir, y se comentan dos de las arquitecturas existentes para el diseño de sistemas multiagente de tiempo real: ObjectAgent y SIMBA.

1. Introducción

En los últimos años, se ha hecho patente la necesidad de dar respuesta a problemas con restricciones de tiempo real de una forma más flexible, adaptativa y orientada al conocimiento que la que se ofrecía con los sistemas de tiempo real clásicos. Dentro del campo de investigación de la inteligencia artificial, los sistemas multiagente [1] han demostrado ser una tecnología útil para afrontar problemas complejos cumpliendo estos requisitos de flexibilidad, adaptabilidad e inteligencia. Por tanto, parece lógico plantearse el uso del paradigma de los sistemas multiagente para afrontar problemas de este tipo en entornos de tiempo real críticos.

Sin embargo, la mayoría de las estrategias distribuidas de resolución de problemas mediante agentes se basan en protocolos de comunicación que no son aplicables a escenarios de tiempo real, ya que los retardos producidos por los procesos de negociación en ocasiones no son predecibles. Por otro lado, la mayoría de las plataformas multiagente que existen en la actualidad han sido diseñadas para satisfacer los requisitos de autonomía, sociabilidad, inteligencia, etc., propios del paradigma de agentes, y no los requisitos de retardos acotados y determinismo propios de un sistema de tiempo real.

Existen diferentes líneas de investigación que pretenden aplicar la tecnología de sistemas multiagente a sistemas de tiempo real, tanto críticos como acríicos. Algunas de ellas disponen de implementaciones en funcionamiento para determinadas aplicaciones. Este trabajo pretende estudiar algunas de las diferentes aproximaciones existentes a los sistemas multiagente de tiempo real, así como las estrategias en las que se basan.

2. Dominios sociales de tiempo real

La mayoría de los sistemas de tiempo real (STR) operan en entornos relativamente simples y bien definidos. A menudo el entorno consiste en un conjunto de tareas cuyo tiempo de ejecución se conoce con exactitud, y cuyo patrón de invocación es conocido. Sin embargo, ciertas aplicaciones complejas que conllevan la utilización de recursos limitados no permiten conocer con exactitud determinista el funcionamiento del sistema en todos los casos. Esto exige al sistema la capacidad de adaptarse a las condiciones del entorno controlado. Esta adaptabilidad puede lograrse mediante la convergencia en un mismo desarrollo de dos de las áreas de investigación en informática: los sistemas de tiempo real y los sistemas inteligentes. Es lo que se conoce como inteligencia artificial en tiempo real (RTAI, *Real Time Artificial Intelligence*) [2]. Si, además, las entidades que componen el sistema se organizan siguiendo una arquitectura distribuida y dependen de la comunicación para llevar a cabo sus objetivos, se habla de inteligencia artificial distribuida en tiempo real o de *dominios sociales de tiempo real* [3].

Existen multitud de escenarios donde se hace necesario un compromiso entre características de tiempo real y requisitos de flexibilidad, adaptabilidad y distribución, como pueden ser ciertos sistemas de comercio electrónico [4], edificios inteligentes [5], sistemas de detección de intrusiones [6] o sistemas de control de tráfico en redes de comunicaciones [7].

2.1. Inteligencia artificial en tiempo real

Tradicionalmente, las técnicas de inteligencia artificial (IA) no se han empleado en entornos de tiempo real debido a que sus tiempos de respuesta suelen ser impredecibles. Esta impredecibilidad es una consecuencia del tipo de problemas en que se han centrado las investigaciones sobre inteligencia artificial, en su mayoría problemas de gran dificultad cuya resolución a menudo implica estrategias de búsqueda dentro de un espacio de soluciones. Una de las tendencias principales en las investigaciones acerca de la aplicación de inteligencia artificial a entornos de tiempo real es la posibilidad de garantizar que el sistema dará siempre una solución válida antes del cumplimiento del *deadline*. La solución tendrá mayor calidad cuanto mayor sea el tiempo disponible para hallarla, de un modo similar a como los seres humanos realizamos razonamientos más impulsivos bajo la presión de restricciones temporales. Es lo que se conoce como técnicas y algoritmos de procesamiento aproximado. Ejemplos de este tipo de técnicas pueden ser el razonamiento progresivo, las búsquedas en tiempo real, la computación imprecisa, los algoritmos *anytime* y los métodos múltiples [8].

Los algoritmos *anytime* son algoritmos de mejora iterativa. Primero se genera una primera solución por defecto, que luego va siendo refinada a través de sucesivas iteraciones. La calidad de la solución hallada se incrementa de forma proporcional al tiempo de que dispone el algoritmo para ejecutarse, pero el algoritmo siempre proporciona una solución, independientemente de cuando sea interrumpido. La aproximación por métodos múltiples se basa en la existencia de un conjunto de métodos para resolver una tarea, cada uno de los cuales posee

diferentes características que lo hacen más o menos apropiado para las condiciones de ejecución. Cada método resuelve el mismo problema, pero difiere en la cantidad de tiempo que tarda en resolverlo y en la calidad de la solución obtenida, lo que permite alcanzar una solución de compromiso entre calidad y tiempo, según las limitaciones que imponga el entorno para un determinado instante.

Las arquitecturas comúnmente empleadas para aplicar técnicas de inteligencia artificial a dominios de tiempo real pueden englobarse en cuatro grandes grupos, en función de cómo realizan la integración entre las capacidades de razonamiento y las restricciones de tiempo real:

- **Arquitecturas que integran inteligencia en un STR.** Este enfoque consiste en añadir comportamiento inteligente a un sistema de tiempo real, de forma que se exige que las tareas de inteligencia artificial cumplan también las restricciones temporales. Pueden emplearse para ello diferentes aproximaciones, como son simplificar los mecanismos de inteligencia artificial hasta que cumplan las restricciones, distinguir entre tareas de tiempo real y tareas de inteligencia artificial y garantizar únicamente el tiempo de ejecución de las primeras, o utilizar algoritmos de mejora iterativa.
- **Arquitecturas reactivas.** En esta aproximación se realizan las inferencias en la fase de diseño del sistema, y no en tiempo de ejecución. Dado el sistema y sus algoritmos de inteligencia artificial, se estudian las posibles entradas al sistema y se diseñan una serie de comportamientos, cada uno responsable de reconocer y reaccionar ante una entrada específica. No pueden ser considerados sistemas inteligentes, ya que carecen de capacidad de inferencia en tiempo de ejecución y su comportamiento es totalmente determinista. Son más bien sistemas de control clásicos de alta complejidad.
- **Arquitecturas que integran reactividad en sistemas de IA.** Estos sistemas utilizan ciertos mecanismos para distinguir las tareas reactivas, de forma que tengan prioridad sobre los procesos convencionales y por lo tanto tengan tiempos de respuesta menores. Sin embargo, si no existe una arquitectura de tiempo real subyacente, no es posible garantizar un tiempo de respuesta acotado.
- **Arquitecturas cooperativas.** Los sistemas que mejor resultado están dando en cuanto a cumplimiento de las restricciones de tiempo real y satisfacción de los objetivos de alto nivel son los que se basan en la cooperación de un subsistema de tiempo real y un módulo de inteligencia artificial, generalmente en procesadores separados. El subsistema de tiempo real es el que interacciona con el entorno a través de los sensores y actuadores (Fig. 1), y garantiza los tiempos de respuesta de todas sus tareas. La responsabilidad de añadir o eliminar tareas del sistema de tiempo real o de modificar su planificación corre a cargo de un planificador, que coopera con el subsistema de inteligencia artificial. Un ejemplo de este tipo de sistemas es la arquitectura CIRCA [9].

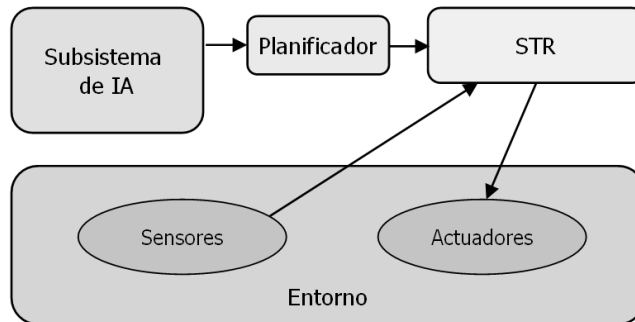


Fig. 1. Esquema de una arquitectura cooperativa para un sistema de inteligencia artificial en tiempo real

3. Sistemas multiagente de tiempo real

La Tecnología de Agentes surgió como un área especializada dentro de la investigación sobre Inteligencia Artificial. Por ello, la mayoría de las investigaciones realizadas en el campo de los sistemas multiagente se centran en diseñar arquitecturas que permitan esquemas de negociación complejos y resolución de tareas de alto nivel, y no en proporcionar sistemas que se adapten a unas determinadas restricciones temporales. En general, las arquitecturas multiagente asumen un canal de comunicación fiable y, aunque algunas permiten el establecimiento de *deadlines* para los procesos de interacción entre agentes, no proporcionan medios que permitan acotar el tiempo que puede llevar al sistema reaccionar ante determinado evento.

De acuerdo con [10], podemos definir un agente de tiempo real como un agente con restricciones temporales en alguna de sus responsabilidades o tareas. A partir de esta definición, puede definirse un sistema multiagente de tiempo real (*Real Time Multi-Agent System*, RT-MAS) como un sistema multiagente en el que al menos un agente es un agente de tiempo real.

El uso de RT-MAS tiene sentido dentro de entornos con restricciones temporales críticas, donde el sistema pueda ser controlado por agentes autónomos, y donde dichos agentes necesiten comunicarse entre sí para mejorar el grado de cumplimiento de los objetivos del sistema. En este tipo de entornos, los agentes necesitan actuar de forma autónoma, pero avanzando hacia un objetivo común. Además, es necesario garantizar la respuesta del sistema en tiempo real, y debe evitarse el establecimiento de comunicaciones excesivamente voluminosas entre agentes. En concreto, para que un sistema multiagente sea aplicable a dominios sociales de tiempo real, debe cumplir las siguientes características [3]:

- Existe un equipo de agentes autónomos A que colabora para lograr un determinado objetivo común a largo plazo (G).
- Periódicamente, cada agente puede enviar o recibir mensajes sin que ello tenga efectos adversos sobre la consecución de G .

- El dominio es dinámico y limitado en el tiempo. Esto significa que la consecución del objetivo G puede verse seriamente afectada si un agente no actúa en un periodo limitado de tiempo.
- Cada agente es capaz de trabajar con restricciones de tiempo real estricto. Esto implica que los tiempos de respuesta de todas las tareas críticas para el sistema deben estar perfectamente acotados.
- El dominio tiene comunicación no fiable, ya sea en cuanto a fiabilidad de la transmisión o a límites de ancho de banda. En definitiva, enviar un mensaje no garantiza su recepción.
- Un mensaje m tiene una longitud máxima K .
- Un agente puede enviar o recibir al menos un mensaje m con un periodo p .
- Si un mensaje m es recibido dentro de las capacidades de recepción especificadas en el punto anterior, será atendido en un intervalo de tiempo acotado.

Las diferentes plataformas cumplen en mayor o menor grado cada uno de estos requisitos, y los afrontan desde diferentes estrategias. A continuación veremos las características específicas de dos de las arquitecturas existentes para sistemas multiagente de tiempo real.

3.1. Arquitectura RT-MAS sobre un Sistema Operativo de Tiempo Real

El sistema ObjectAgent [11] fue desarrollado por Princeton Satellite Systems para crear una arquitectura basada en agentes para sistemas autónomos distribuidos que pudiera ser aplicable a sistemas de *clusters* de satélites. Uno de los objetivos principales era emplear agentes que se comunicaran entre sí mediante mensajes sencillos para implementar toda la funcionalidad software del sistema de satélites. Otro punto clave del diseño era incorporar funcionalidad de toma de decisiones, tolerancia a fallos y recuperación tanto a alto como bajo nivel.

La implementación embarcable de ObjectAgent se desarrolló para el sistema operativo de tiempo real OSE de Enea [12], elegido por sus características de control de recursos por procesos, comunicación a través de señales, manejo de errores jerárquico y actualización dinámica de software, que lo hace muy adecuado para la aplicación considerada.

ObjectAgent utiliza una planificación por prioridades con desalojo que le permite mejorar los tiempos de respuesta de las tareas de más alta prioridad. Sin embargo, es responsabilidad del diseñador acotar los tiempos de respuesta de las tareas de inteligencia artificial para garantizar los tiempos de respuesta de las tareas de menor prioridad. Si el sistema no se diseña cuidadosamente, una tarea de inteligencia artificial de prioridad alta puede entrar en un ciclo de búsqueda de alta varianza e impedir la ejecución de tareas menos prioritarias durante un tiempo impredecible, afectando al determinismo del sistema. Asimismo, los mensajes en ObjectAgent representan las tareas que el sistema debe realizar, y el procesado de los mensajes sigue una política FIFO, por lo que el orden de llegada de los mensajes puede dar lugar a inversiones de prioridad. Por último,

objetivos de una forma aceptable. Por otro lado, puesto que debe existir una máquina con dos procesadores por cada agente ARTIS existente en el sistema, la escalabilidad de la arquitectura limita su campo de aplicación. Como aspecto particularmente interesante de la planificación en SIMBA está el uso de técnicas de robo de holgura (*Slack Stealing*)[15] para compaginar la ejecución de tareas críticas y la ejecución de tareas de inteligencia artificial y comunicaciones, que no tienen restricciones temporales estrictas.

4. Conclusiones

Existe un creciente interés en acercar las disciplinas de inteligencia artificial y de sistemas de tiempo real, motivado principalmente por las enormes de posibilidades de aplicación que ofrece la inteligencia artificial en tiempo real en sectores tan dispares como la domótica, el control de procesos industriales o la administración de redes de comunicaciones. Muchas de estas aplicaciones se beneficiarían además de un enfoque distribuido para la solución al problema. Constituyen lo que se conoce como dominios sociales de tiempo real, y sugieren el uso de sistemas multiagente como una tecnología bastante prometedora para su desarrollo. Sin embargo, la utilización de sistemas multiagente plantea una dificultad añadida a las ya señaladas, ya que la comunicación entre entidades distribuidas es un proceso no determinista y no fiable, cuyos tiempos de respuesta no pueden acotarse con facilidad.

Existen varias líneas de trabajo que proponen diferentes arquitecturas y metodologías para el desarrollo de sistemas multiagente de tiempo real. Aunque parten de las mismas premisas generales, las estrategias que emplean y la importancia que confieren a cada uno de los requisitos de los entornos sociales de tiempo real son muy distintas, por lo que también las características finales de la arquitectura difieren considerablemente.

No existe en la actualidad una solución general para la construcción de sistemas aplicables a dominios sociales de tiempo real. En su lugar, existen diferentes soluciones que abordan el problema desde un punto de vista particular, dando mayor prioridad a unos u otros requisitos según el campo de al que vaya dirigido el sistema. A la hora de afrontar un problema concreto para una aplicación determinada, deben valorarse cuidadosamente los requisitos del problema y las restricciones impuestas por el entorno, a fin de determinar la arquitectura más adecuada para el sistema que lo resuelva.

Agradecimientos

Este trabajo ha sido realizado gracias a la financiación del Ministerio de Ciencia y Tecnología, a través del proyecto MCYT-TIC2003-09192-C11-05, así como de la Universidad de Alcalá, mediante el proyecto UAH-PI-2003/001.

Referencias

1. Jennings, N., Wooldridge, M.: Software agents. *IEE Review* (1996) 17–20
2. Weiss, R., Steger, C.: Design and implementation of a real-time multi-agent system. In: 9th Mediterranean Electrotechnical Conference, *melecon '98*, Tel-Aviv, Israel (1998) 1269–1273
3. Julian, V., Carrascosa, C., Rebollo, M., Soler, J., Botti, V.: Simba: an approach for real-time multi-agent systems. In: *Proceedings of V Conferencia Catalana d'Intelligencia Artificial*, Castelló, Springer-Verlag (2002)
4. DiPippo, L.C., Wolfe, V.F., Nair, L., Hodys, E., Uvarov, O.: A real-time multi-agent system architecture for e-commerce applications. Technical Report TR00-280, University of Rhode Island (2000)
5. Cook, D.J., Youngblood, M., Edwin O. Heierman, I., Gopalratnam, K., Rao, S., Litvin, A., Khawaja, F.: Mavhome: An agent-based smart home. In: *First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, Fort Worth, Texas (2003)
6. Gorodetski, V., Kotenko, I., Karsaev, O.: Multi-agent technologies for computer network security: Attack simulation, intrusion detection and intrusion detection learning. *Internacional Journal of Computer Science and Engineering* **18** (2003) 191–200
7. Jennings, B., Brennan, R., Gustavsson, R., Feldt, R., Pitt, J., Prouskas, K., Quantz, J.: Fipa-compliant agents for real-time control of intelligent network traffic. *Computer Networks*, **31** (1999) 2017–2036
8. Hodys, E.: A scheduling algorithm for a real-time multi-agent system. Master's thesis, University of Rhode Island (2000)
9. Musliner, D.J., Durfee, E.H., Shin, K.G.: Circa: A cooperative intelligent realtime control architecture. *IEEE Trans. Systems, Man, and Cybernetics* **23** (1993)
10. Julian, V., Botti, V.: Developing real-time multi-agent systems. In: *Proceedings of the Fourth Iberoamerican Workshop on Multi-Agent Systems (Iberagents'02)*, Málaga (2004)
11. Surka, D.M., Brito, M., Harvey, C.G.: The real-time objectagent software architecture for distributed satellite systems. In: *2001 IEEE Aerospace Conference Proceedings*, Big Sky, Montana (2001)
12. Enea OSE's Webpage, <http://www.ose.com>.
13. Soler, J., Julian, V., Rebollo, M., Carrascosa, C., Botti, V.: Towards a real-time multi-agent system architecture. In: *In Proceedings of the First Internacional Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, Bologna, Italy (2002)
14. Botti, V., Carrascosa, C., Julian, V., Soler, J.: Modelling agents in hard real-time environments. In: *MAAMAW'99 (LNAI Vol. 1647)*, Springer-Verlag (1999) 63–76
15. Davis, R.I., Tindell, K.W., Burns, A.: Scheduling slack time in fixed priority pre-emptive systems. In: *Proceedings of the 14th IEEE Real-Time Systems Symposium*. (1993)