

INTEGRACIÓN DE PLATAFORMAS ROBÓTICAS Y MULTI-AGENTE EN SISTEMAS DE FABRICACIÓN

B. Fortes, A. Armentia, U. Gangoiti, D. Orive, I. Sarachaga, M. Marcos
Dept. Ingeniería de Sistemas y Automática, UPV/EHU, España
e-mail: brais.fortes@gmail.com,

{aintzane.armentia, unai.gangoiti, dario.orive, isabel.sarachaga, marga.marcos}@ehu.eus

Resumen

En los últimos años se ha producido una evolución de los sectores industriales para adaptarse a los importantes cambios en la demanda de productos: mayor personalización, mayor calidad, menor precio y menor tiempo de proceso productivo. Es por ello que las explotaciones modernas demandan un sistema de control cómodo, flexible y descentralizado. En este contexto, los sistemas de transporte se presentan como un factor determinante para poder llevar a cabo la reconfiguración de estos sistemas. Sin embargo, los marcos robóticos empleados para su desarrollo carecen de las capacidades sociales necesarias para que varios robots puedan interactuar. El presente trabajo propone una arquitectura para integrar sistemas robóticos con sistemas multi-agente. Dicha arquitectura no sólo resuelve la interacción entre robots sino que también transforma dichos sistemas de transporte en sistemas de transporte inteligentes dotados con capacidad de toma de decisiones. La arquitectura de integración propuesta se ha probado en un caso de estudio relativo a un sistema de fabricación flexible.

Palabras clave: sistemas de transporte inteligente; sistemas de fabricación flexible; JADE; ROS; arquitectura de integración

1 INTRODUCCIÓN

A lo largo de los últimos años, el sector secundario, en general, y la tecnología, en particular, han avanzado mucho. Estos cambios no han sido aislados dentro del sector industrial, sino que en parte han sido promovidos por los cambios de la sociedad y el mercado. Esto se puede traducir en términos económicos como el paso de un mercado local y controlado a un mercado completamente globalizado, al que tienen acceso incluso las pequeñas y medianas empresas (PYMEs); ya sea de un modo directo a partir de sus productos, principalmente tecnológicos, o indirecto como proveedores de grandes exportadores.

Esta evolución continua ha implicado cambios importantes en la demanda de los productos. Mientras los procesos industriales tradicionales, centralizados y jerárquicos, son rentables para procesos de fabricación estáticos, con redundancias de máquinas ante fallos y lotes de fabricación muy grandes; estos no se adaptan en absoluto a las exigencias actuales de un mercado que pide mayor personalización, mayor calidad, menor precio y menor tiempo en los procesos productivos [10].

Para dar solución a esta problemática, los sectores industriales ya hablan desde hace algunos años de la Industria 4.0. Un sistema en el cual un conjunto de elementos heterogéneos (sensores, máquinas, sistemas de TI, etc.) se conectan a la cadena de valor más allá de la propia empresa. Los sistemas se comunican entre sí, recopilando y analizando datos entre máquinas, aumentando su autonomía, la velocidad de los procesos, la flexibilidad y, en definitiva, la eficiencia [6].

Aunque la Industria 4.0 contempla la conexión de toda la cadena de valor, los sistemas de integración de los procesos productivos, en sí mismos, proponen dotar a la fábrica de mecanismos para aumentar la sencillez, modularidad y eficiencia, trasladando el control centralizado del proceso a los propios productos, máquinas CNC, robots autoguiados (*Automated Guided Vehicle, AGV*) u operarios, entre otros muchos. Estos sistemas de fabricación son llamados en la literatura como sistemas de fabricación orientados a productos (*Product Oriented Manufacturing*) o, simplemente, sistemas de fabricación flexible (*Flexible Manufacturing System*) [7], [9].

Las necesidades de descentralizar los procesos de fabricación dotando de autonomía a todos y cada uno de los componentes que forman el proceso son pues, evidentes. Es por ello que las plantas industriales deben incorporar nuevas tecnologías con técnicas de inteligencia artificial que permitan la cooperación y distribución de decisiones dentro de sus cualidades. Las explotaciones modernas necesitan un control cómodo y flexible a la hora de trabajar y obtener beneficios. En este contexto, los sistemas de

transporte son uno de los factores más determinantes para poder llevar a cabo la reconfiguración de estos sistemas, especialmente en el caso de las líneas de producción.

En concreto, el uso de robots móviles es una manera de implementar un sistema de transporte flexible. No obstante, las relaciones sociales, imprescindibles para la toma de decisiones descentralizadas, no tienen una solución trivial dentro del desarrollo de robots, ya que las plataformas robóticas no han incluido estas capacidades dentro de sus características intrínsecas. Es por ello que la implementación de estas nuevas características en sistemas robóticos de transporte, debe tender hacia entornos de trabajo ya probados y fiables como pueden ser algunos de los frameworks de sistemas multi-agente (*Multi-Agent System, MAS*).

Los sistemas MAS han sido objeto de investigación desde hace ya algunos años, y aunque su implementación real dentro de los procesos de fabricación no es demasiado elevada, su potencialidad es enorme a la hora de flexibilizar los sistemas productivos, llegando a considerarse uno de los sistemas clave en la fabricación avanzada [3].

Este trabajo propone un sistema de entendimiento entre dos marcos completamente independientes, como son el robótico y el multi-agente. De este modo, es posible implementar de forma sencilla cualquier sistema robótico con características modulares, robustas, eficientes y tolerantes ante fallos. La arquitectura de integración descrita en este trabajo añade versatilidad frente a las habituales soluciones *ad hoc*, permitiendo su reutilización y extensión a campos de aplicación diferentes de los proyectados en este artículo.

El resto de este trabajo está estructurado de la siguiente manera: la sección 2 define los requisitos no funcionales del sistema de transporte inteligente y, por lo tanto, de la arquitectura de integración; la sección 3 identifica las posibilidades que ofrecen los marcos robóticos y multi-agente en la creación de sistemas multi-robot. La sección 4, muestra la solución de integración propuesta y los mecanismos necesarios para ponerla en marcha. La sección 5 está dedicada al caso de estudio orientado a la fabricación, dotando de inteligencia al sistema de transporte. Por último, en la sección 6 se resumen las conclusiones más importantes y el trabajo futuro.

2 IDENTIFICACIÓN DE REQUISITOS

Un sistema de fabricación inteligente está formado por operarios, almacenes, máquinas, células, líneas de producción, estaciones de carga para AGVs y robots que realizan un conjunto de operaciones con el

objetivo de gestionar un plan en un plazo determinado.

En este contexto, el objetivo principal que deben cumplir estos transportes consiste en el movimiento de materia prima, productos procesados o finales dentro del área de trabajo de un modo eficiente. Es decir, evitando obstáculos y ofreciendo el servicio del transporte con aquel robot que sea capaz de gestionarlo más rápido.

Por lo tanto, aunque es habitual que las máquinas y los pedidos basen su actividad en un plan inicial, que se adapta con el tiempo ante las necesidades del proceso productivo, los AGVs son ligeramente diferentes, ya que no siguen ningún orden de montaje. Deben responder a las necesidades de las máquinas y de los pedidos en ejecución, solucionando cualquier adversidad física o lógica que se haya producido en la planta.

En este sentido, durante la ejecución normal del proceso productivo, cualquier robot de la planta debe cooperar y/o interactuar con otros elementos del proceso para realizar sus tareas. De este modo, el sistema se compone de elementos heterogéneos, distribuidos, interconectados entre sí, debiendo ser capaces de tomar decisiones por sí mismos.

Teniendo todo esto en cuenta, la Tabla 1 recoge los requisitos identificados para un sistema de transporte inteligente y que son descritos con más detalle en los siguientes párrafos.

Tabla 1 Requisitos del transporte inteligente.

ID	Descripción
R1	Soporte y gestión de sensores y actuadores.
R2	Capacidad social o negociadora que le permite comunicarse con otros elementos del sistema.
R3	Toma de decisiones racional y autónoma.
R4	Comunicación con sistemas robóticos y no robóticos, independientemente de sistema operativo, hardware o tipo.
R5	Conexión fluida entre equipos, sin retrasos comunicativos.
R6	Reconfiguración ante cambios de contexto externos o internos.
R7	Recuperación frente a fallos de robot.

Los AGVs que operan sobre una misma fábrica tienen características técnicas diversas, siendo necesario gestionar elementos heterogéneos dentro del mismo equipo robótico (R1). Además, los procesos de fabricación flexible obligan a comunicar elementos del proceso productivo para cooperar y/o

negociar, tomando decisiones conjuntas gracias al conocimiento parcial del sistema de cada uno de sus elementos (R2, R3, R4). No es suficiente con comunicar robots entre ellos, sino que es necesario procesar también las peticiones externas de máquinas y pedidos.

Así pues, es muy importante no saturar la red con información redundante o innecesaria. Una planta industrial no admitiría la pérdida o el retraso de mensajes con plazos muy cortos debido a una mala gestión de las comunicaciones (R5).

Por último, los sistemas de transporte inteligente también deben responder ante situaciones anómalas y de fallo, debiendo ser capaces de recalcular su trayectoria ante obstáculos en el camino, de gestionar peticiones preferentes o detectar fallos internos (R5, R6).

3 ANÁLISIS DE LOS FRAMEWORKS EMPLEADOS EN SISTEMAS MULTI-ROBOT

Esta sección presenta las características y necesidades básicas de los frameworks habitualmente empleados para el desarrollo de entornos robóticos (apartado 3.1) y entornos MAS (apartado 3.2), los dos marcos a integrar en el presente trabajo. Esta visión general, permite reconocer las carencias y fortalezas de estos sistemas para cumplir con los requisitos indicados en el apartado 2. Y por lo tanto, la necesidad de desarrollar una arquitectura para su integración.

3.1 MARCOS ROBÓTICOS

Los avances tecnológicos han hecho que en la última década los robots sean cada vez más vistos en entornos industriales e, incluso, domésticos. Esto ha permitido que diversos grupos de desarrollo internacionales hayan implementado marcos robóticos (ROS, YARP, OROCOS, ORCA, etc.) que dotan de modularidad, fiabilidad y simplicidad al desarrollo de nuevos robots. La intención de estas plataformas es abstraer a los desarrolladores de todas las problemáticas relativas al hardware (dependiente de cada robot), para permitir desarrollar nuevas funcionalidades independientemente de quién sea el responsable físico de ejecutarlas. Por lo tanto, el uso de cualquiera de estas plataforma asegura el cumplimiento del requisito R1 (ver Tabla 1). Sin embargo, todos estos frameworks robóticos no disponen de una arquitectura social que les permita establecer resoluciones cooperativas, sobre todo con elementos no robóticos (requisitos R2 y R3).

Este trabajo ha decidido analizar y poner en valor las características del marco robótico *open source* más

utilizado en el ámbito de la investigación, ROS (*Robotic Operating System*). Destaca frente a otros frameworks del mismo tipo debido a que se basa en una arquitectura distribuida, completamente modular, y dispone de una gran comunidad de usuarios, extensa documentación y un amplio repositorio de librerías y drivers. Además, también soporta la programación en múltiples lenguajes de programación (C++, Python, Octave, LISP, Java, etc.), facilitando las tareas de los desarrolladores.

Su diseño arquitectónico modular se basa en dos componentes principales: nodos, unidad mínima de procesamiento del sistema; y nodo Master, registro y lugar de búsqueda de información. El lanzamiento de una aplicación robótica implica pues, la creación de múltiples nodos internos del robot que intercambian información y razonan para realizar la acción necesaria.

Los intercambios de información pueden gestionarse a partir de diferentes criterios: publicador/suscriptor, cliente/servidor y/o acciones. Todas estas comunicaciones entre nodos ROS se sustentan por los protocolos de comunicaciones habituales (TCP/IP, UDP/IP, etc.) [5].

En este contexto, se ha analizado en qué medida un sistema de transporte inteligente desarrollado totalmente en ROS permite cumplir con los requisitos de la Tabla 1. Al igual que el resto de plataformas robóticas asegura el soporte y gestión de elementos hardware heterogéneos (R1), pero no integra las características sociales dentro de la arquitectura. En el caso concreto de ROS, existen paquetes externos (ROS multimaster [11]) que permiten comunicar diferentes elementos robóticos que empleen este marco, dotando al sistema de ciertas capacidades sociales y autónomas (R2, R3).

El mayor problema surge a la hora de comunicar el sistema de transporte con el resto del proceso productivo. ROS multimaster permite relacionar robots con robots, pero no robots con elementos no robóticos (R4). Además, si se analiza un poco más en detalle el funcionamiento de este paquete, es sencillo deducir que las comunicaciones que establece entre robots son muy agresivas, con demasiados mensajes, existiendo riesgo de saturar la red (requisito R5).

3.2 MARCOS MULTI-AGENTE

Los sistemas MAS son una rama de la Inteligencia Artificial Distribuida que trata la resolución de problemas de forma cooperativa a partir de entidades, denominadas agentes [8]. Sus objetivos principales pasan por resolver problemas complejos a partir de la subdivisión de tareas sencillas. Para ello, los agentes

deben ser autónomos, sociales, racionales, móviles, pro-activos, y adaptables.

Este trabajo ha decidido utilizar un framework multi-agente que se rija por metodologías estándar que doten de interoperabilidad al sistema MAS. De este modo, se utiliza la plataforma JADE, que implementa completamente el estándar FIPA (*Foundation for Intelligent Physical Agents*). Su funcionamiento es sencillo, obligando a tener tres agentes por defecto: DF (*Directory Facilitator*), encargado de almacenar los servicios de los agentes; AMS (*Agent Management System*), gestiona la creación, eliminación y creación de agentes; y el ACC (*Agent Communication Channel*) que soporta la movilidad entre agentes. Una vez los mecanismos de gestión están en el sistema, es posible incorporar nuevos agentes al MAS. Los intercambios de información se realizan a partir de mensajes ACL (*Agent Communication Language*) definidos por el propio estándar para realizar interacciones ligeras y entendibles por todos los agentes.

Conocida esta información, es posible analizar si un sistema MAS se adapta a las necesidades previamente identificadas. Las características intrínsecas de este tipo de sistemas orientados al intercambio de información, hacen que los requisitos R2, R3, R4 y R5 se cumplan sin necesidad de ningún desarrollo añadido. Además, los requisitos R6 y R7, pese a que no son propios de la arquitectura multi-agente, sí pueden ser solucionados mediante plataformas que basándose en frameworks MAS (como es el caso de JADE) añaden los módulos necesarios [1]. No obstante, el requisito R1, fuertemente ligado a las necesidades hardware de los sistemas robóticos, no tiene fácil solución desde el enfoque multi-agente.

Teniendo en cuenta todo lo anterior se concluye que la incorporación de un sistema multi-agente propiamente dicho a los frameworks robóticos supone un avance a la hora de implementar la socialización de estos robots [2]. Sin embargo, para que ambos marcos cooperen es necesario definir una arquitectura de integración, como la detallada en el apartado 4, que permita, por un lado, solucionar los problemas de comunicaciones, y por otro, permita independizar el trabajo de dichos marcos, permitiendo que especialistas en diferentes áreas puedan trabajar conjuntamente.

4 ARQUITECTURA DE INTEGRACIÓN

El objetivo principal de la arquitectura de integración propuesta es compatibilizar todas las características funcionales de bajo nivel que ofrecen los frameworks robóticos con las capacidades sociales propias de los

sistemas multi-agente. De este modo, la conexión con el hardware y todos los procesos relativos al mismo es implementado en el marco robótico, mientras que el MAS se ocupa de las relaciones de alto nivel de comunicación entre robots y con el sistema de fabricación del que forman parte.

Además, la solución propuesta no pretende romper los criterios de arquitectura de ninguna de las dos plataformas. Es por ello que los nuevos nodos no se dotarán de una gran carga de trabajo, teniendo en cuenta que están pensados para realizar tareas simples; y los nuevos agentes tampoco tendrán que procesar mucha información, ya que retrasarían en gran medida su ciclo de ejecución.

Se propone una arquitectura genérica de cuatro capas (Figura 1) que da simplicidad y claridad a cada uno de los procesos del sistema. Su intención es separar responsabilidades para aumentar la solidez, modularidad y eficiencia del sistema. En este sentido, permite que cambios en los nodos no supongan ningún cambio significativo para el sistema.

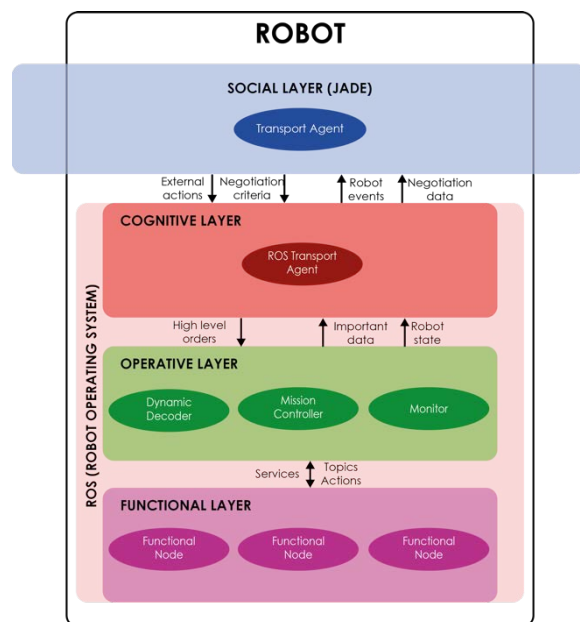


Figura 1: Arquitectura de integración ROS-JADE

En rasgos muy generales, cada robot perteneciente al sistema está compuesto por un agente MAS que le permite comunicarse y decidir de forma autónoma (capa social); y por un conjunto de nodos ROS que controlan la integración multi-capa y la funcionalidad del robot (capa funcional). La comunicación entre ambos sistemas se realiza a partir de un objeto ROS (capa cognitiva) instanciado por la capa social. De este modo, el agente MAS puede comunicarse con los nodos ROS de bajo nivel a partir de los métodos de la capa cognitiva, y viceversa.

Como los nodos funcionales de ROS son reutilizables y modulares, se añade una capa de abstracción superior a la funcionalidad (capa operativa), para comunicarse con el agente MAS y entender sus peticiones.

A continuación se describe con detalle cada una de estas capas.

4.1 Capa social

Esta capa consta del llamado agente transporte, agente JADE que engloba todas las acciones de alto nivel, abstrayéndose completamente del posible nodo físico sobre el que se ejecute. Su función principal es la socialización, es decir, la comunicación con otros elementos del sistema MAS en el que se integra. En este sentido, se ocupa de atender a las peticiones de otros agentes del sistema, entrar en negociaciones o proporcionar información actual del estado.

Además, esta capa social también tiene cierta inteligencia y autonomía para tomar decisiones, ya que es la encargada de decidir que función de alto nivel debe ejecutar el robot en cada momento. Así pues, cada vez que un robot deba modificar la operación que está realizando, el agente JADE debe notificárselo por comunicaciones.

Aunque la arquitectura expuesta define un único agente por robot dentro de la capa social, existe la posibilidad de emplear más si alguna aplicación determinada lo requiriese.

4.2 Capa cognitiva

Es la capa que hace de enlace entre el sistema JADE y ROS. Aunque funcionalmente no tiene valor añadido para el sistema, es la capa más importante de la integración, ya que debe ser transparente para dos plataformas completamente diferentes.

Los servicios principales que implementa este nodo (agente transporte ROS) engloban acciones como (1) el envío de información de la capa social a la capa cognitiva, y viceversa; (2) el almacenamiento de información relevante para el sistema o (3) la respuesta a peticiones de información particulares de la capa social. Se trata de servicios sencillos que no realizan ningún tipo de preprocesado de la información que almacenan o envían mediante mensajes.

Los métodos de comunicación vertical empleados por esta capa son dependientes de la dirección en la que se realicen. Así pues, para comunicarse con las capas operativas emplea los métodos implementados por el marco robótico ROS (tópicos, servicios y

acciones). Mientras que la comunicación con el agente transporte es directa debido a instancias.

Su implementación se ha apoyado en la librería ROSJava, con la intención de utilizar el mismo lenguaje de programación que su capa superior. Esto permite disponer de un nodo ROS asíncrono, debido a las características intrínsecas de la librería para mejorar el rendimiento y la cohesión.

4.3 Capa operativa

Es la capa que se ocupa de traducir las órdenes de más alto nivel en órdenes entendibles por los nodos funcionales de más bajo nivel, y viceversa. Aunque los nodos ROS que la conforman están fuertemente ligados a los servicios que ofrece el robot en el sistema MAS, el hecho de emplear nodos sencillos, clasificados por tipos permite la reutilización funcional de estas unidades mínimas de procesamiento. Así, todas las comunicaciones e implementaciones de esta capa siguen los estándares de ROS.

A continuación se describen brevemente los nodos que la conforman:

1. Mission Controller: Se ocupa de mantener la máquina de estados de un robot determinado, ejecutando y controlando las órdenes de alto nivel. Su implementación se basa en el análisis y subdivisión de las tareas abstractas recibidas de sus capas superiores y en el procesamiento de las mismas. Además, también realimenta a las capas superiores con el estado de las operaciones que han solicitado. Facilitando que tomen decisiones en caso de que ocurran problemas.
2. Monitores: Leen y solicitan toda la información de la red robótica que es de interés para la socialización del robot. Una vez obtenida, la analizan, pre-procesan y envían a partir de tópicos a la capa cognitiva. Este procedimiento asegura que toda la información de relevancia sea conocida por el cerebro del robot, presente en las capas de abstracción superiores (capa social).

Además, estos monitores realizan tareas de filtrado de información que puede no ser de interés, evitando saturar las comunicaciones debido a su alta frecuencia.
3. Decodificadores Dinámicos: Traducen directamente solicitudes de los agentes de alto nivel. De este modo, implementan funcionalidades diversas como: gestión de operaciones no esperadas (fallo, modificación de criterios, etc.) o comunicación interna dentro de

la capa operativa (solicitudes de estados o cambios).

Este tipo de nodos, al poder comunicarse mediante ROS con los elementos de su misma capa, hace que la arquitectura de la capa operativa sea completamente modular, abstrayendo al mision controller y a los monitores de este tipo de gestiones.

5 CASO DE ESTUDIO

Con el objetivo de validar la implementación, se ha introducido la arquitectura propuesta en un sistema de fabricación flexible basado en agentes (Figura 2). Su diseño distribuido extiende las características presentadas en [4], distinguiendo tres tipos de agentes en el sistema productivo: agentes de supervisión (SA y MoA), agentes de producción (OA y BA) y agentes recurso (MA y TA).

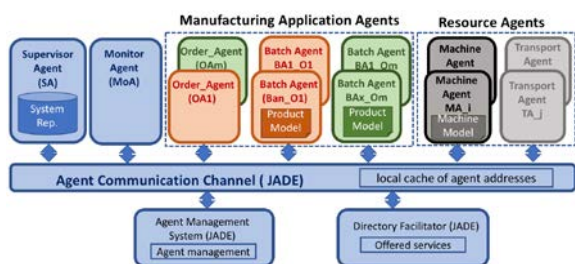


Figura 2. Arquitectura MAS de sistema de fabricación flexible.

El agente supervisor (*Supervisor Agent*, SA) registra todos los elementos del sistema, sus servicios y su estado. Por tanto, contiene la información más actualizada del sistema en su conjunto, solucionando así problemas de recuperación de fallos y atención a cambios de contexto. De esta forma se cumple con los requisitos R6 y R7 (Tabla 1) del transporte inteligente.

El otro agente de supervisión *Monitor Agent* (MoA) permite introducir a los operarios en el sistema MAS. Por un lado, monitoriza cierta información del sistema y, por otro, permite realizar peticiones u órdenes de operario al sistema.

Los agentes de producción *Order Agent* (OA) y *Batch Agent* (BA) son los encargados de hacer el seguimiento del proceso de fabricación. Es decir, almacenan y gestionan toda la información relativa a los pasos necesarios para la fabricación de un producto. En concreto, los OA hacen un seguimiento de diversos BA y, estos últimos, llevan la trazabilidad del producto a partir de su interacción con las máquinas (*Machine Agent*, MA).

Finalmente, los agentes más próximos a los sistemas robóticos son los recursos: MA y *Transport Agent* (TA). Estos son representantes de elementos físicos y funcionales del proceso de fabricación. En este sentido, los MA representan a las máquinas de un proceso productivo, implementando diferentes servicios (taladrado, montaje, etc.); y los TA son los AGVs que mueven material por la planta industrial.

La integración JADE-ROS es una solución óptima para integrar los sistemas robóticos de transporte del proceso de fabricación.

Así, debe definirse un agente transporte en su capa social con información e inteligencia suficiente para poder realizar sus tareas. En el caso de estudio, los agentes transporte ofrecen tres tipos de servicio: (1) reposición/reparto: mover material de punto a punto; (2) rescate: buscar algún AGV fallido, coger su carga y terminar su tarea; (3) exploración: buscar un objeto u obstáculo concreto en la planta industrial.

Estas tareas no están planificadas al inicio del proceso productivo, por lo que los agentes transporte (TA) deben atender a peticiones de los lotes de productos (BA) y las máquinas (MA). Una vez recibidas, todos los agentes transporte negocian, de modo que aquel TA con mejores prestaciones temporales, incluirá en su plan la nueva tarea.

La capa social, por tanto, debe ser capaz de registrar los servicios propios de su AGV en el sistema, negociar la realización de tareas y gestionar su plan dinámico. Para ello, el agente tiene que poder acceder a información local de bajo nivel del robot y debe ser capaz de dar órdenes a los actuadores, es decir, de ordenar la realización de nuevas tareas.

La arquitectura de integración emplea una capa cognitiva que es completamente transparente para el sistema, abstrayendo así a los TA de la implementación de la funcionalidad del robot. Es la capa operativa la que define nodos ROS propios a la aplicación en estudio. A este nivel, existe (1) una máquina de estados (mission controller) que desgrana las órdenes de alto nivel y las gestiona con los nodos funcionales; (2) diversos monitores que analizan la información relevante para las negociaciones (posición en la planta industrial, batería o distancia entre puntos); y (3) diversos decodificadores dinámicos, preparados para atender a peticiones no esperadas en operación normal (por ejemplo, parada de emergencia).

En definitiva, se comprueba que la arquitectura de integración JADE-ROS se adapta perfectamente a las necesidades del transporte inteligente en un proceso industrial.

6 CONCLUSIONES Y TRABAJO FUTURO

Este trabajo presenta una solución genérica para integrar el marco robótico ROS y el sistema multi-agente JADE. La integración permite que diversos robots cooperen y compitan así como que dichos robots interactúen de forma sencilla con cualquier elemento de un mismo sistema MAS. En este sentido, se puede concluir que aporta los mecanismos necesarios para implementar capacidades sociales en sistemas robóticos de las que de por sí mismos carecen. Además, permite mantener equipos de desarrollo separados según las áreas de conocimiento existentes implicadas en la integración (robótica y multi-agente). De este modo, no es necesario ser experto de ambos sistemas para implementar un robot con capacidades sociales.

La arquitectura de integración se ha empleado en un sistema de fabricación flexible basado en agentes. De esta manera se ha podido probar que, en lo que a la fabricación respecta, la solución propuesta permite resolver uno de los aspectos más relevantes del sistema productivo: el transporte inteligente. Dando soporte a las necesidades de las máquinas en situación normal y en situaciones de re-planificación de la producción, por ejemplo debido a la caída de alguna máquina o a la llegada de lotes de alta prioridad.

El trabajo futuro se centra en probar la arquitectura propuesta en un entorno real dotado de robots móviles. Así mismo, se pretende simplificar a partir de plantillas el uso de esta arquitectura.

Agradecimientos

Este trabajo se ha subvencionado en parte por el Gobierno de España bajo el proyecto DPI2015-68602-R (MINECO/FEDER, UE), por la UPV/EHU bajo el proyecto PPG17/56 y por el GV/EJ bajo el grupo de investigación reconocido IT914-16.

English summary

INTEGRATION OF ROBOTIC AND MULTI-AGENT PLATFORMS IN MANUFACTURING SYSTEMS

Abstract

During the last years the industrial sector has evolved in order to adapt to changes on products demand: more personalization, higher quality, lower prices and lower time-to-market. As a result modern industry demands an easier as well as more flexible

and decentralized control system. In this context, the transport systems seem a key factor to achieve the reconfiguration of those systems. However, the robotic frameworks used for developing such transports lack of the social capacities needed for the interaction among robots. Current work contributes an architecture that integrates robotic systems with multi-agent systems. Thus, the integration not only solves the interaction issue but also it introduces intelligence in the transport system, being capable of taking decisions. The proposed integration architecture has been proved in a case study related to a flexible manufacturing system.

Keywords: Intelligent transport system; flexible manufacturing system; JADE; ROS; integration architecture.

Referencias

- [1] Armentia, A., Gangoit, U., Priego, R., Estévez, E., Marcos, M. (2015). *Flexibility support for homecare applications based on models and multi-agent technology*. Sensors, Vol: 15, Issue: 12, pp. 31939-31964.
- [2] Iñigo-Blasco, P., Diaz-del-Rio, F., Romero-Ternero, M. C., Cagigas-Muñiz, D., Vicente-Diaz, S. (2012). *Robotics software frameworks for multi-agent robotic systems development*. Robotics and Autonomous Systems, Vol: 60, Issue: 6, pp. 803-821.
- [3] Leitão, P. (2009). *Agent-based distributed manufacturing control: A state-of-the-art survey*. Engineering Applications of Artificial Intelligence, Vol: 22, Issue: 7, pp. 979-991.
- [4] López M., Martín J., Gangoit, U., Armentia A., Estévez E., Marcos M. (Accepted for publication). *Supporting product oriented manufacturing: a Model Driven and Agent based Approach*. In Proceedings of the IEEE 16th International Conference of Industrial Informatics (INDIN 2018).
- [5] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Ng, A. Y. (2009). *ROS: an open-source Robot Operating System*. In Proceedings of the ICRA workshop on open source software, Vol: 3, No: 3.2, pp. 5.
- [6] Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., Harnisch, M. (2015). *Industry 4.0: The future of productivity and growth in manufacturing industries*. Boston Consulting Group.

- [7] Silva, S. C., Alves, A. (2002). *Design of Product Oriented Manufacturing Systems. In Knowledge and Technology integration in production and Services*. In Proceedings of the International Conference on Information Technology for Balanced Automation Systems, pp. 359-366.
- [8] Shoham, Y. (1993). *Agent-oriented programming*. Artificial intelligence, Vol: 60, Issue: 1, pp. 51-92.
- [9] Stecke, K. E. (1985). Design, planning, scheduling, and control problems of flexible manufacturing systems. *Annals of Operations research*, Vol: 3, Issue: 1, pp. 1-12.
- [10] Storey, J. (1994). *New wave manufacturing strategies: Organizational and human resource management dimensions*. Paul Chapman Publishing, pp. 272.
- [11] Tiderko, A., Hoeller, F., Röhling, T. (2016). *The ROS multimaster extension for simplified deployment of multi-robot systems*. Robot Operating System (ROS), pp. 629-650.



© 2018 by the authors.
Submitted for possible
open access publication
under the terms and conditions of the Creative
Commons Attribution CC-BY-NC 3.0 license
(<https://creativecommons.org/licenses/by-nc/3.0>).