

COMPONENTE I4.0 DE UNA ESTACIÓN DE ENSAMBLADO

A. Aragón, A. López, I. Sarachaga, O. Casquero, D. Orive, A. Armentia, F. Pérez
 Departamento de Ingeniería de Sistemas y Automática UPV/EHU
 aaragon006@ikasle.ehu.es, {alejandro.lopez, isabel.sarachaga, oskar.casquero, dario.orive,
 aintzane.armentia, federico.perez}@ehu.es

Resumen

El concepto de fábrica inteligente puede empezar a cimentarse mediante una red de Sistemas Ciber-Físicos (CPS) interconectados. La abstracción y/o virtualización de los activos de fabricación en cuanto a sus funcionalidades, proporcionadas como servicios, es una condición necesaria para materializar el concepto de CPS. Éste es el enfoque seguido por RAMI 4.0, en el que el concepto del Componente I4.0 juega un papel fundamental para convertir cualquier activo de fabricación en un CPS. Los agentes industriales constituyen una solución adecuada para implementar Componentes I4.0. En este sentido, existen arquitecturas y prácticas recomendadas que permiten articular metodologías para la implementación de Componentes I4.0 mediante agentes. Este trabajo describe cómo se pueden llevar a la práctica esas metodologías mediante la propuesta de acciones concretas para la implementación de un Componente I4.0 de una estación de ensamblado integrada en una plataforma de gestión de aplicaciones de fabricación basada en agentes.

Palabras clave: Componente I4.0, agente industrial, sistema multi-agente de fabricación flexible.

1 INTRODUCCIÓN

Los sistemas de fabricación convencionales se encuentran inmersos en una transición hacia modelos de producción inteligentes y personalizables, capaces de asegurar una gestión eficiente de los recursos de fabricación, y de adaptar su producción dinámicamente a las demandas actuales del mercado y a las incidencias en planta.

El paradigma de la Industria 4.0 plantea el uso de herramientas y tecnologías que aumenten la eficiencia y la capacidad de adaptación de los procesos productivos, en aras de una fabricación inteligente y descentralizada [13]. Muchos gobiernos y empresas han considerado este paradigma como factor clave en sus estrategias de desarrollo industrial, dando lugar a iniciativas como Industrie 4.0 en Alemania e Industrial Internet en Estados Unidos, entre otras [4].

En general, todas ellas conciben la fabricación inteligente como la interconexión entre activos de fabricación, cuya virtualización proporciona las funcionalidades como servicios de red que permiten una comunicación vertical y horizontal.

Concretamente, la arquitectura de referencia RAMI 4.0 (Reference Architectural Model Industry 4.0) contempla el Componente I4.0 como una descripción virtual de un objeto con conexión digital para interoperar dentro de un sistema I4.0 [5]. Un Componente I4.0 comprende el *activo (asset)*, un objeto físico o lógico con valor para la empresa), y un *administrador del activo (Asset Administration Shell – AAS)* que proporciona acceso digital a la funcionalidad técnica del objeto virtual mediante *submodelos* de información a los que se puede acceder a través de un *gestor de componentes* [15], [20].

Los Componentes I4.0 deben cumplir ciertos requisitos: proporcionar capacidades de comunicación para garantizar la interoperabilidad entre los componentes I4.0 del sistema; identificar de forma exclusiva tanto el AAS como el activo; representar el estado y los servicios del activo en submodelos; gestionar el acceso de otros componentes a los submodelos del componente I4.0; integrar los AAS y los activos; y, por último, gestionar la automatización y ejecución de los servicios realizados por el activo. Una alternativa para la implementación de los AAS apunta a los agentes software, que son entidades autónomas capaces de competir o colaborar con otros agentes del sistema para alcanzar sus objetivos [16]. Dichos agentes permiten satisfacer todos los requisitos asociados a los Componentes I4.0, a excepción de los correspondientes a la integración y gestión del activo.

El paradigma de agentes industriales surge para hacer frente a esta limitación. Sin embargo, la escalabilidad y reutilización de soluciones basadas en agentes industriales siguen constituyendo un reto debido a la diversidad de activos físicos presentes en una fábrica en cuanto a la heterogeneidad de las interfaces de comunicación de los activos físicos y la ubicación de los agentes. Para abordar este problema, se han propuesto dos soluciones metodológicas. Por una parte, el estándar IEEE 2660.1-2020 define unas prácticas recomendadas que identifican cuatro

escenarios de despliegue para la integración de un agente con un activo físico [6]. Por otra parte, la arquitectura de capas descrita en [10] proporciona un marco para encapsular los diferentes aspectos de la integración en diferentes niveles de abstracción.

Partiendo de estas soluciones metodológicas, este trabajo identifica las acciones concretas para la implementación de un Componente I4.0 de una estación de ensamblado (formada por un robot colaborativo UR3e y automatizada con un PLC de la gama S7-1500) como un agente industrial en el contexto de la plataforma de gestión de aplicaciones de fabricación basada en agentes FlexManSys [9]. Esta integración comprende: la identificación y programación de las operaciones de la estación de ensamblado de forma modular para maximizar la flexibilidad del sistema; la definición de una interfaz, a nivel de operación, para el intercambio de datos entre el activo y su agente asociado; y la definición de una interfaz, a nivel de servicio, que permita la gestión virtual de la estación.

La estructura del artículo es la siguiente: en el apartado 2 se presenta una visión general de la plataforma de gestión de aplicaciones de fabricación FlexManSys, y de la arquitectura de capas y las prácticas recomendadas para la integración genérica de AAS basados en agentes con activos físicos. En el apartado 3 se detalla la solución propuesta para la integración del activo de fabricación (estación de ensamblado) como componente I4.0 en FlexManSys, mientras que el apartado 4 describe el proceso de depuración de dicho componente I4.0. Finalmente, la sección 5 destaca las principales aportaciones y el trabajo futuro a corto plazo.

2 ANTECEDENTES

Con objeto de establecer el contexto de la integración del Componente I4.0 de la estación de ensamblado en la plataforma de gestión de aplicaciones de fabricación se precisan unas nociones básicas acerca de dicha plataforma, así como sobre la arquitectura de capas y las prácticas recomendadas que se utilizan como guías para la integración.

FlexManSys [9] se fundamenta en un middleware basado en agentes cuyo núcleo consiste en el *Agente del Sistema*, que proporciona una interfaz para dar de alta, iniciar y detener aplicaciones de fabricación, así como consultar y actualizar el estado del sistema de fabricación (almacenado en el *Modelo del Sistema*) durante todo el ciclo de ejecución de la aplicación. La trazabilidad de las aplicaciones de fabricación se realiza a diferentes niveles mediante los *Agentes de Planes de Manufactura*, *Agentes de Pedidos* y *Agentes*

de Lote [11]. Además, los *Agentes Recurso* representan los servicios y necesidades de los recursos disponibles en la fábrica, encargados de interactuar con su activo, representando su estado y ofreciendo sus servicios al resto de agentes del sistema cuando entra en negociación para realizar operaciones. Se contemplan tres tipos de agentes recurso: *Agentes Máquina*, que ofrecen servicios de fabricación, *Agentes de Transporte*, que ofrecen servicios de transporte, y *Agentes de Nodo de Procesamiento*, que ofrecen servicios de procesamiento de datos.

En este trabajo se realiza una implementación del *Agente Máquina* como agente industrial conforme a la arquitectura de cuatro capas descrita en [10]: una capa superior o de inteligencia, que aglutina las capacidades de socialización y la toma de decisiones asociadas a los agentes; una capa intermedia o de gestión, que transforma los datos recibidos del activo en información útil para los Componentes I4.0 en el sistema; y dos capas inferiores, operativa y funcional, que gestionan la llamada, la ejecución y el estado de los servicios del activo.

Esta arquitectura de capas es compatible con el estándar IEEE 2660.1-2020, ya que mientras la arquitectura se centra en el encapsulado de los diferentes aspectos necesarios para implementar el Componente I4.0, el estándar se centra en la interacción entre agentes y activos. Concretamente, el estándar presenta cuatro prácticas de integración, atendiendo al acoplamiento entre el agente y el activo, así como al despliegue del agente: el acoplamiento puede ser fuerte (conexión directa y permanente) o débil (conexión orientada a datos), mientras que el despliegue puede ser in situ (el agente se despliega en el propio activo físico) o híbrido (el agente se despliega fuera del activo físico).

3 ESCENARIO DE INTEGRACIÓN

La solución propuesta para la integración del activo de fabricación (estación de ensamblado) en un sistema de fabricación flexible, conforme al paradigma Industria 4.0, se realiza integrando dicho activo de fabricación en FlexManSys como un Componente I4.0.

En la plataforma FlexManSys, todas las máquinas de una planta (assets físicos) están gestionadas por su correspondiente agente máquina (AAS basado en agentes) capaz de interactuar con el resto de agentes de la plataforma para tomar decisiones sobre los servicios a realizar por la máquina. Dicha interacción se produce en forma de negociación entre agentes máquina para repartirse la asignación de tarea, siendo el agente del sistema quien inicia la negociación entre los agentes máquina. Por su parte, la toma de

decisiones requiere que el agente máquina conozca el estado de las tareas que su asset físico tiene en curso.

El asset físico es el responsable de realizar los servicios de fabricación que la plataforma inteligente considere necesarios en cada momento, así como de informar acerca de la evolución de las tareas solicitadas. El asset físico se corresponde con una estación de ensamblado formada por un robot colaborativo UR3e que está automatizada con un PLC de la gama S7-1500. El robot es el encargado de realizar los servicios de fabricación relacionados con el ensamblaje de piezas.

En este contexto, la Figura 1 muestra la relación del Componente I4.0 con la abstracción por aspectos funcionales según la arquitectura de capas, y con la interacción entre el agente y el activo según el estándar IEEE 2660.1-2020. En lo que se refiere a la arquitectura de capas, el agente máquina (AAS) representa las capas superiores de la arquitectura de integración (capas de inteligencia y gestión), mientras que la estación de ensamblado (asset, dentro del cual se incluyen el robot y el PLC), se corresponde con las capas inferiores (capas operativa y funcional). En lo que se refiere al despliegue del agente y el activo, en este caso se opta por un acoplamiento débil y un despliegue híbrido.

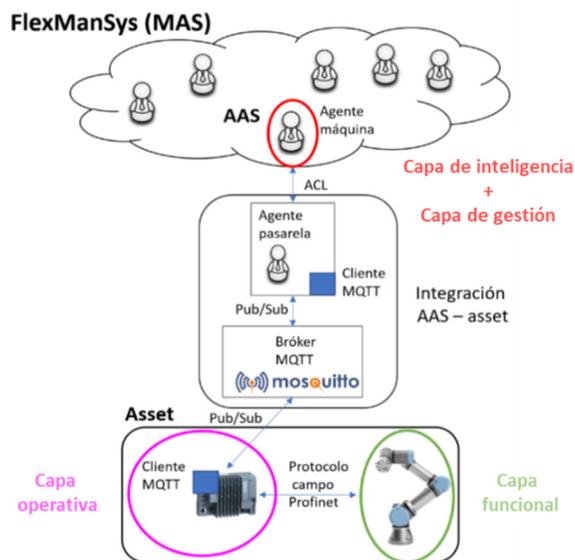


Figura 1: Escenario de integración

La capa de inteligencia es la encargada de gestionar las interacciones con otros agentes y procesar las peticiones de servicio, mientras que la capa de gestión se encarga de gestionar los submodelos del activo. Por tanto, estas dos primeras capas están directamente relacionadas con el agente máquina. En las capas definidas en el activo de fabricación se diferencia entre: por un lado, la capa operativa que se sitúa en el PLC, encargada de recibir las peticiones y realizar la

llamada al correspondiente programa del robot parametrizándolo; y, por otro lado, la capa funcional asociada al robot colaborativo UR3e, donde se efectúan las operaciones solicitadas.

La integración de los activos de fabricación en agentes industriales no es directa, siendo necesaria una interfaz para la integración AAS-asset. Esta interfaz debe adaptar los mecanismos de comunicación propios de los agentes a las interfaces de comunicación de la estación de ensamblado. Con el objetivo de desacoplar al máximo los dos extremos de la comunicación (AAS y asset), este trabajo se caracteriza por un acoplamiento débil como resultado del uso de una comunicación orientada a datos basada en el protocolo MQTT [12] y un despliegue híbrido en el que el agente se despliega fuera del asset, que en un entorno de producción podría ser un nodo de procesamiento.

Por último, cabe destacar que el agente máquina, como entidad software de FlexManSys, es una plantilla que define la interfaz, los submodelos y los comportamientos comunes a todas las máquinas en una fábrica. Es decir, los agentes máquina concretos asociados a máquinas concretas son instancias de dicha plantilla. Para que dicha plantilla sea independiente de la interfaz de comunicación del asset, se introduce un *Agente Pasarela*, que actúa de puerta de enlace entre el agente máquina y su asset, convirtiendo mensajes ACL (Agent Communication Language) propios de un sistema multi-agente a mensajes del tipo de protocolo definido en la interfaz del asset (en este caso, MQTT).

3.1 AAS DEL COMPONENTE I4.0

La implementación del agente máquina se ha realizado en el entorno de desarrollo integrado IntelliJ IDEA [7] utilizando Java como lenguaje de programación. También se ha requerido JADE (Java Agent Development Framework) [8] como apoyo al desarrollo de sistemas multi-agente, que proporciona una implementación en Java compatible con las especificaciones FIPA (Foundation Intelligent Physical Agents) [2].

La estructura de los mensajes que se intercambian los agentes es conforme al lenguaje ACL también definido por FIPA [3]. Los mensajes son de confianza, fiables y ordenados, es decir, los mensajes están bien formados y llegan correctamente a su destino consiguiendo que los mensajes se reciban tal y como se envían. En este proyecto se han empleado tres tipos de mensajes: REQUEST para solicitar la ejecución de una acción desde el agente al asset, CONFIRM para ratificar la validez de una solicitud, e INFORM para reportar estado desde el asset.

Todos los agentes siguen la misma secuencia de arranque, en la que se inicializan y son registrados en el sistema antes de realizar las tareas que se les haya asignado. En el caso de los agentes que gestionan la aplicación (*Agentes de Planes de Manufactura, Agentes de Pedidos y Agentes de Lote*), mueren una vez han completado su cometido en el desarrollo de la aplicación. Sin embargo, en el caso del agente máquina, su ciclo de vida no está vinculado a una aplicación concreta, por lo que nunca muere (tiene un comportamiento cíclico).

Una vez definido el comportamiento del agente, se le debe asignar una tarea. Para ello, el agente máquina, recopila la información necesaria para poder comenzar con el servicio de ensamblaje a cargo del robot colaborativo. Dicha información se guarda en una estructura de datos definida como un HashMap, permitiendo proporcionar la información de manera ordenada para realizar la comunicación vertical hacia el asset. El agente pasarela recibe dicha estructura de datos en forma de solicitud (REQUEST), tras lo cual envía confirmación (CONFIRM) al agente máquina y éste queda bloqueado hasta que los mensajes de estado que indican la evolución del proceso (INFORM) lleguen desde el agente pasarela.

3.2 ASSET DEL COMPONENTE I4.0

El asset de la estación de ensamblado consta de un PLC de la gama S7-1500 y el robot colaborativo UR3e. El robot se configura como esclavo Profinet del PLC [14].

El proyecto de automatización del PLC se ha desarrollado en TIA Portal V16 [19], usando librerías que facilitan la comunicación MQTT y la manipulación de las estructuras de datos relacionadas con el intercambio de datos entre agente y asset, y entre PLC y robot.

El robot colaborativo está programado para realizar el ensamblado de ítems que constan de las siguientes piezas: base, rodamiento, bulón, tapa interior y tapa exterior (Figura 2). Las bases sobre las cuales se ensamblan las piezas vienen ordenadas en un palet con 6 posiciones.

La mesa de trabajo está organizada en 5 zonas identificadas por colores (Figura 3). La zona de entrada (zona A en verde) es donde se sitúa el palet al inicio del servicio. Al comienzo de cada ensamblaje, el robot coge el palet situado en esta zona y lo lleva a la zona de montaje (zona B en rojo). Una vez finalizado el montaje de todos los ítems, el robot transporta el palet a la zona de salida (zona C en negro), donde se depositan los palets finalizados. En color amarillo se identifica el dispensador de bulones, en color morado el dispensador de rodamientos y en

color naranja el dispensador de tapas interiores. También se puede apreciar la zona de entrada de tapas exteriores (zona D en azul).

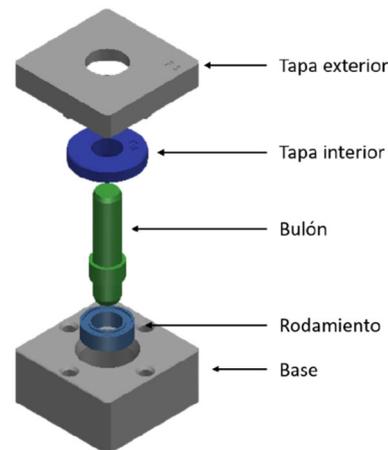


Figura 2: Conjunto a ensamblar

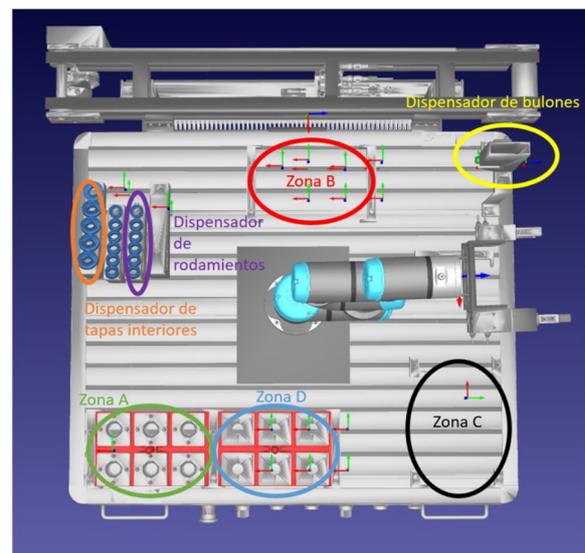


Figura 3: Mesa de trabajo del robot UR3e

La programación del robot sigue los principios de modularidad para dotar de mayor flexibilidad a la estación de ensamblado, tanto en producción normal como en situaciones excepcionales de producción, y permitir la reutilización de las distintas fases en las que se estructuran los 10 servicios. Las fases hacen referencia a los movimientos entre zonas que el robot debe realizar para completar el servicio.

Con la solicitud de servicio procedente del agente máquina, el PLC envía al robot tanto el servicio solicitado como la cantidad de ítems a realizar (que puede variar entre uno y seis). El programa cíclico del robot evalúa los datos y ejecuta unas fases u otras según corresponda al servicio.

Dependiendo del servicio solicitado, el robot realiza un montaje completo sobre la base o coloca las piezas que faltan en montajes de partida incompletos, tal y como se puede apreciar en la Tabla 1.

El palet en la zona de inicio dispondrá siempre de bases como mínimo, aunque no es necesario trabajar sobre los seis ítems que componen el palet.

3.3 INTEGRACIÓN AAS-ASSET

La integración entre el AAS y el asset precisa de un agente pasarela para realizar la conversión entre el protocolo ACL y el protocolo MQTT. El agente pasarela tiene un comportamiento cíclico y su tarea consiste en enviar la estructura de datos procedente del agente máquina al bróker MQTT y viceversa. Cuando recibe una estructura de datos procedente del agente máquina, el agente pasarela se encarga de enviar un mensaje de confirmación para asegurar la correcta recepción, procesa dichos datos y realiza el envío de la estructura de datos al bróker MQTT. Cuando los datos provienen del bróker, el agente pasarela se encarga de procesarlos y enviarlos al agente máquina mediante mensajes informativos.

Tabla 1: Servicios proporcionados por el asset.

Servicio	Montaje inicial	Piezas a montar
1	Base	Rodamiento
2		Rodamiento Bulón
3		Rodamiento Bulón Tapa interior
4		Rodamiento Bulón Tapa interior Tapa exterior
5	Base + Rodamiento	Bulón
6		Bulón Tapa interior
7		Bulón Tapa interior Tapa exterior
8	Base + Rodamiento + Bulón	Tapa interior
9		Tapa interior Tapa exterior
10	Base + Rodamiento + Bulón + Tapa interior	Tapa exterior

La comunicación MQTT entre el agente pasarela y el PLC se realiza a través de los tópicos /UR3e/Agent2Asset y /UR3/Asset2Agent mediante sendos clientes MQTT que publican datos en y se suscriben a dichos tópicos. Concretamente, el agente

pasarela precisa de un cliente MQTT encargado de publicar los datos procedentes del agente máquina en el tópico /UR3e/Agent2Asset y de suscribirse al tópico /UR3/Asset2Agent para recibir la información de estado procedente del asset. Este cliente se ha implementado haciendo uso de la librería Eclipse Paho Java Client. Por otro lado, el proyecto de automatización del PLC dispone de un bloque de función (FB) que implementa un cliente MQTT. Este FB publica la información de estado procedente del asset en el tópico /UR3/Asset2Agent y se suscribe al tópico /UR3e/Agent2Asset para recibir solicitudes de servicio procedentes del agente máquina. Este cliente se ha implementado haciendo uso de la librería LMQTT para controladores SIMATIC.

Por último, hay que destacar la utilización de Eclipse Mosquitto [1] como bróker MQTT por ser un proyecto de código abierto adecuado para cualquier tipo de dispositivo debido a su ligereza.

4 ESCENARIO DE PRUEBAS

La depuración del componente I4.0 ha requerido el escenario de la Figura 4 para la realización de: 1) pruebas unitarias de los dispositivos y de conexión entre dispositivos con objeto de validar las tecnologías empleadas; 2) pruebas de integración con estrategia bottom-up para comprobar que la comunicación vertical en ambos sentidos proporciona una transferencia de datos correcta; así como 3) pruebas de verificación y validación del componente I4.0 de la estación de ensamblado.

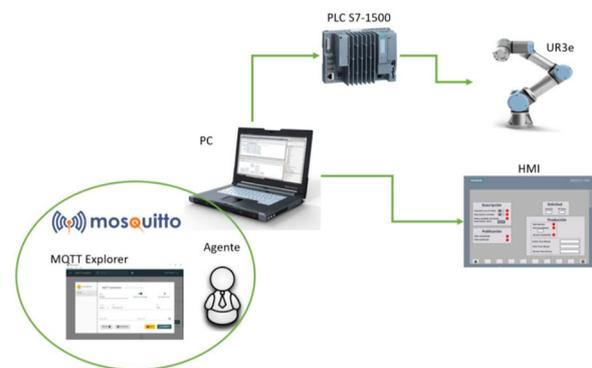


Figura 4: Escenario de pruebas

Como se puede observar, en el escenario de pruebas intervienen el PLC y el robot UR3e (el PLC automatiza la ejecución de los servicios programados en la controladora del robot UR3e), y las aplicaciones que se ejecutan en el PC local. Concretamente, estas aplicaciones son las siguientes: los agentes máquina y pasarela, el software MQTT Explorer que permite visualizar los datos que llegan al bróker en ambos sentidos, y el HMI que permite controlar las variables del proceso y mostrar su evolución.

4.1 PRUEBAS UNITARIAS

Las pruebas unitarias del robot están directamente relacionadas con la comprobación de la correcta programación de los movimientos del robot. Conocida la distribución de la mesa, se ha comprobado que el robot llega adecuadamente y sin colisiones a cada una de las posiciones requeridas.

Las pruebas de conexión entre PLC y robot consisten en el envío de una sola variable en ambos sentidos de la comunicación para asegurar la correcta recepción del dato. Por su parte, las pruebas de conexión entre PLC y bróker consisten en el envío de estructuras de datos serializadas en JSON en ambos sentidos.

Mientras que las pruebas de conexión entre el PLC y el bróker han podido realizarse mediante simulación con la herramienta PLCSim Advanced [18], las pruebas correspondientes a la conexión entre PLC y robot se han realizado con el PLC físico, dada la imposibilidad de simular conexiones Profinet mediante PLCSim Advanced.

Las pruebas correspondientes a la conexión entre el agente máquina y el agente pasarela se han realizado con ayuda de una consola que permite escribir manualmente la variable a enviar desde el agente máquina al agente pasarela. Tras la correcta recepción, el agente pasarela envía un mensaje de confirmación.

Por último, las pruebas de conexión entre el bróker y el agente pasarela han involucrado al bróker, al agente pasarela y al agente máquina, siendo arrancados en este orden. Así, la variable introducida en la consola del agente máquina llega al agente pasarela y su cliente MQTT la publica para que llegue al bróker. En sentido contrario, mediante la consola que ofrece MQTT Explorer se ha realizado el envío de una variable desde el bróker al tópicos en el que el cliente MQTT del agente pasarela está suscrito para recibirla.

4.2 PRUEBAS DE INTEGRACIÓN

Las pruebas de integración se han realizado según una estrategia bottom-up. En primer lugar, se ha procedido a la integración de los dispositivos que conforman el asset. Una vez obtenida una comunicación correcta en ambos sentidos entre el PLC y el robot, el siguiente paso ha sido añadir el bróker. Para finalizar con las pruebas de integración se han agregado los agentes pasarela y máquina.

Cabe destacar que, si bien las pruebas unitarias se ciñen a la conexión entre dispositivos, las pruebas de integración se focalizan en el correcto intercambio de las estructuras de datos asociadas tanto a las solicitudes de servicios como a la información sobre el desarrollo de dichos servicios.

Las pruebas de integración iniciales del asset han consistido en el intercambio de datos booleanos entre los dos dispositivos. El PLC activa un bit que se corresponde con la variable MARCHA, con objeto de arrancar el programa del robot. Por su parte, el robot envía dos bits diferentes: uno cada vez que finaliza el ensamblaje de un ítem y otro cuando finaliza el proceso completo y el palet está en la zona de salida de la mesa de trabajo. Las siguientes pruebas han añadido el envío de valores enteros para definir el servicio que se quiere realizar y la cantidad de ítems.

Para realizar las pruebas de integración del bróker con el asset se ha usado la consola que ofrece MQTT Explorer, en la cual se han definido todos los datos del proceso que deben llegar hasta la controladora del robot de manera ordenada y se ha visualizado la información que llega al bróker desde el asset. También se ha requerido la librería LStream (Library for Data Streams) [17], la cual ofrece una serie de bloques que facilitan la conversión de la matriz de bytes en formato de datos JSON procedente del bróker a un formato que pueda ser procesado por un controlador SIMATIC S7, y viceversa.

Por último, se han añadido el agente máquina y el agente pasarela y se ha probado la solicitud de un servicio, introduciendo los siguientes datos por consola: referencia de la máquina, referencia del pedido, referencia del lote, referencia del tipo del subproducto, tipo de servicio a realizar y cantidad de ítems.

4.3 VERIFICACIÓN Y VALIDACIÓN

Las pruebas de verificación y validación se han realizado con ayuda de un HMI (Figura 5) organizado en cuatro secciones: *suscripción* (arriba a la izquierda) con dos interruptores que permiten la conexión y la suscripción del cliente MQTT al bróker con sus correspondientes indicadores de color de aviso; *publicación* (abajo a la izquierda) con dos indicadores de color que señalan el momento en que se serializan los datos que se envían al bróker y el instante en el que se realiza la publicación; *solicitud* (arriba a la derecha) donde se muestra el servicio y el número de ítems solicitados; y *producción* (abajo a la derecha) donde se muestran los datos actualizados de producción.

5 CONCLUSIONES

El trabajo presenta la integración de un activo de fabricación (una estación de ensamblado) como componente I4.0 en un sistema multi-agente de fabricación flexible, indicando las acciones necesarias para la implementación de dicho componente.



Figura 5: HMI de pruebas

El robot colaborativo UR3e se centra en la realización de los servicios de ensamblaje de piezas desde diferentes montajes iniciales (base, base con rodamiento, etc.) y con diferente número de ítems, así como en el envío de la información relativa a la evolución del servicio. La estructura modular del programa del robot ha permitido aumentar la eficiencia y flexibilidad del proceso de ensamblado, tanto en producción normal como en situaciones excepcionales de producción.

El PLC es el encargado de atender las solicitudes del agente máquina, haciéndolas llegar al robot, y del tratamiento de la información recibida del robot para convertirla a la estructura demandada por el agente. Además, el hecho de disponer de un PLC también ha permitido ampliar las capacidades de comunicación propias del robot, permitiendo usar el paradigma publicista/suscriptor para la integración AAS-asset, siendo el bróker Mosquitto y el agente pasarela los intermediarios de dicha comunicación.

Como trabajo futuro a corto plazo se contempla la integración total del componente I4.0 en la arquitectura de FlexManSys, de tal manera que no sea necesario el uso de la consola en el agente máquina. Así, la selección del servicio se realizará mediante la negociación entre agentes, siendo la solicitud del servicio seleccionado una de las acciones derivadas de dicha negociación.

Agradecimientos

Este trabajo está financiado por MCIU/AEI/FEDER, UE (proyecto RTI2018-096116-B-I00) y por UPV/EHU (proyecto PES18/48).

English summary

I4.0 COMPONENT OF AN ASSEMBLY STATION

Abstract

The smart factory concept can start to be built by a network of interconnected Cyber-Physical Systems (CPS). The abstraction and/or virtualization of manufacturing assets in relation with their functionalities, provided as services, is a necessary condition to materialize the CPS concept. This is the RAMI 4.0 approach, in which the I4.0 Component concept plays a key role in order to transform a manufacturing asset into a CPS. Industrial agents are a suitable solution to implement I4.0 Components. In this sense, there are architectures and best practices that allow articulating methodologies for implementing I4.0 Components using agents. This paper describes how these methodologies can be put into practice by proposing concrete actions for the implementation of an I4.0 Component of an assembly station, which has been integrated in an agent-based manufacturing application management platform.

Keywords: I4.0 component, industrial agent, flexible multi-agent manufacturing system.

Referencias

- [1] “Eclipse Mosquitto”, <https://mosquitto.org/> (acceso en junio de 2022).
- [2] “FIPA specifications”, (2002) <http://www.fipa.org/repository/standardspecs.html> (acceso en junio de 2022).
- [3] “FIPA ACL Message Structure Specification”, (2002) <http://www.fipa.org/specs/fipa00061/SC00061G.html> (acceso en junio de 2022).
- [4] Fraile, F., Sanchis, R., Ortiz, A., (2019) “Reference Models for Digital Manufacturing Platforms”, Applied Sciences, vol. 9, no. 20, pp. 4433.
- [5] IEC, (2017) “IEC PAS 63088: Smart Manufacturing – Reference Architecture Model Industry 4.0 (RAMI4.0)”, Publicly Available Specification, Pre-Standard.
- [6] IEEE Std 2660.1-2020, (2021) “IEEE Recommended Practice for Industrial Agents: Integration of Software Agents and Low-Level

- Automation Functions", pp.1-43, DOI: 10.1109/IEEEESTD.2021.9340089.
- [7] "IntelliJ IDEA: el IDE de Java eficaz y ergonómico de JetBrains", <https://www.jetbrains.com/es-es/idea/> (acceso en junio de 2022).
- [8] "JADE (JAVA Agent DEvelopment Framework)", <https://jade.tilab.com/> (acceso en junio de 2022).
- [9] López, A., Estevez, E., Casquero, O., Marcos, M., (2020) "Using industrial standards for modeling flexible manufacturing systems," IEEE Conference on Industrial Cyberphysical Systems (ICPS 2020), pp. 41–46. DOI:10.1109/ICPS48405.2020.9274785.
- [10] Lopez, A., Casquero, O., Estevez, E., Leitao, P., Marcos, M., (2021) "Towards the generic integration of agent-based AASs and Physical Assets: a four-layered architecture approach", IEEE 19th International Conference on Industrial Informatics (INDIN), pp. 1–6. DOI: 10.1109/indin45523.2021.9557568.
- [11] López, A., Estévez, E., Marcos, M. Trazabilidad de la producción basada en agentes industriales. En XLII Jornadas de Automática: libro de actas. Castelló, 1-3 de septiembre de 2021 (pp. 717-723). DOI capítulo: <https://doi.org/10.17979>
- [12] Oasis, (2019) "MQTT Version 5.0", <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf> (acceso en junio de 2022).
- [13] Pascual, D.G., Daponte, P., Kumar, U., (2020) "Handbook of Industry 4.0 and SMART Systems", CRC Press.
- [14] PI North America, "PROFINET Technology", <https://us.profinet.com/technology/profinet/> (acceso en junio de 2022).
- [15] Platform Industrie 4.0, (2018) "The structure of the Administration Shell, Trilateral Perspectives from France, Italy and Germany", International paper.
- [16] Seitz, M. Gehlhoff, F, Cruz Salazar, L., Fay, A., Vogel-Heuser, B., (2021) "Automation platform independent multi-agent system for robust networks of production resources in industry 4.0", Journal of Intelligent Manufacturing, n. 32, pp. 1-19. DOI:10.1007/s10845-021-01759-2.
- [17] Siemens, "Library for Data Streams (LStream)", <https://support.industry.siemens.com/cs/ww/en/view/10978116> (acceso en junio de 2022).
- [18] Siemens, "SIMATIC S7-PLCSIM Advanced", [https://support.industry.siemens.com/cs/document/109772889/descarga-del-simatic-s7-plcsim-advanced-v3-0-de-prueba-\(trial\)?dti=0&lc=es-DO](https://support.industry.siemens.com/cs/document/109772889/descarga-del-simatic-s7-plcsim-advanced-v3-0-de-prueba-(trial)?dti=0&lc=es-DO) (acceso en junio de 2022).
- [19] Siemens, "TIA Portal V16", <https://support.industry.siemens.com/cs/document/109771626/liberaci%C3%B3n-para-el-suministro-del-tia-portal-v16?dti=0&lc=es-WW> (acceso en junio de 2022).
- [20] ZVEI, (2020) "Details of the Asset Administration Shell. Part 1 – The exchange of information between partners in the value chain of Industrie 4.0 (Version 2.0)", Specification.



© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).