

Desarrollo de sistemas industriales mediante dispositivos empotrados basados en Java

José Miguel Gutiérrez-Guerrero, Jesús Luis Muros-Cobos, Sandra Rodríguez-Valenzuela, Miguel Damas-Hermoso, Juan Antonio Holgado-Terriza

Resumen—Este artículo explica cómo desarrollar sistemas para automatizar procesos industriales utilizando tecnologías abiertas. Para ello se introducen dos middleware basados en tecnología Java que facilitan el desarrollo de estos sistemas. Por una parte, JavaES (Java for Embedded Systems) se utilizará para interactuar directamente con el hardware, y por otra parte, DOHA (Dynamic Open Home Automation) proporcionará un medio para con los PLCs de los sistemas de control a través de una tecnología orientada a servicios.

Palabras clave—SOA, Servicios, Buses de Campo, PLC, HMI.

I. INTRODUCCIÓN

El mundo de la automatización de los sistemas industriales ha avanzado vertiginosamente a la par del desarrollo tecnológico. De un modo progresivo se han ido incorporando al sector industrial metodologías de diseño que en principio no fueron pensadas para este tipo de sistemas, como por ejemplo las tecnologías orientadas a servicios [1].

Las ventajas que las nuevas tecnologías pueden ofrecer a los sistemas industriales son numerosas. Además, la progresiva evolución de los dispositivos empotrados amplía el número de aplicaciones y de sistemas que se están beneficiando de estos avances, dando cada vez más soporte y sentido a proyectos como Socrades [2]. Sin embargo, la mentalidad conservadora de los sistemas industriales hace que la adopción de estos avances tecnológicos, tan bien asumidos en otros dominios, sea muy difícil o se aplique de forma muy lenta y con muchas restricciones en este ámbito.

Una de las metodologías que está teniendo una mayor aceptación en entornos industriales, por las grandes ventajas que ofrece, es *Service-Oriented Architecture* (SOA), y más concretamente la materialización de esta tecnología mediante *Web Services* (WS) [3]. En este ámbito una de las especificaciones de servicios web más utilizada es *Devices Profile Web Services* (DPWS) o servicios web para dispositivos con pocos recursos [4]. Este estándar especifica los aspectos más relevantes y las restricciones que debe seguir una implementación SOA basada en servicios web para garantizar su funcionamiento en dispositivos con escasos recursos, como dispositivos empotrados y hardware específico. Existen numerosos trabajos en los que se aplica la metodología SOA materializada mediante alguna implementación del estándar DPWS al mundo industrial, pues las posibilidades de aplicación de SOA en ámbitos industriales son enormes [4][5].

En este trabajo vamos a utilizar la plataforma orientada a servicios *Dynamic Open Home-Automation* (DOHA) como plataforma para la monitorización de sistemas industriales basados en servicios web, pudiendo ser utilizada por los sistemas *Human Machine Interface* (HMI) para la monitorización y parametrización de procesos industriales. Además, utilizaremos *Java for Embedded Systems* (JavaES) como framework para interactuar con el hardware industrial, aislándonos de las características específicas del empotrado a utilizar y de la arquitectura empleada para el acceso a los dispositivos industriales. El acceso puede ser directo cuando el envío o recepción de la información se hace sin dispositivos intermedios, o subrogado cuando se utiliza un dispositivo intermedio como pasarela para el envío o recepción de la información. Por otro lado, abordaremos los aspectos relacionados con la comunicación entre dispositivos mediante la utilización de buses de campo en entornos industriales. Proponemos una solución que nos permita trabajar con la gran mayoría de buses de campo existentes.

El artículo está organizado como sigue. En la Sección II se hace un pequeño resumen de los sistemas industriales actuales y de cómo están evolucionando. A continuación, en la Sección III se explican las tecnologías JavaES y DOHA que se van a utilizar en el diseño y desarrollo de sistemas industriales. Utilizando ambas tecnologías, se propone una solución a los distintos aspectos (comunicación PLC-PC [6], interconexión entre buses de campo [7]) de los sistemas industriales en la Sección IV. En la Sección V se presenta un caso de estudio con distintos posibles enfoques arquitectónicos. Finalmente se exponen los trabajos futuros y las conclusiones, secciones VI y VII.

II. SISTEMAS INDUSTRIALES ACTUALES

Las soluciones tradicionales para la automatización de procesos industriales se basan en la utilización de tres elementos principales: buses de campo, *Programmable Logic Controllers* (PLC) y sistemas HMI. La aparición de estos elementos supuso una revolución en el diseño e implantación de sistemas de automatización industriales, si bien su utilización y desarrollo fue cronológicamente distinto [8]. En la actualidad, cualquier proceso, máquina o sistema industrial que se desee automatizar utiliza, o debería utilizar, estos elementos. Por ello, los desarrollos futuros o avances en el campo de la industria, y más en concreto en la automatización de procesos industriales, deben de contemplar y tener en cuenta todos los aspectos que

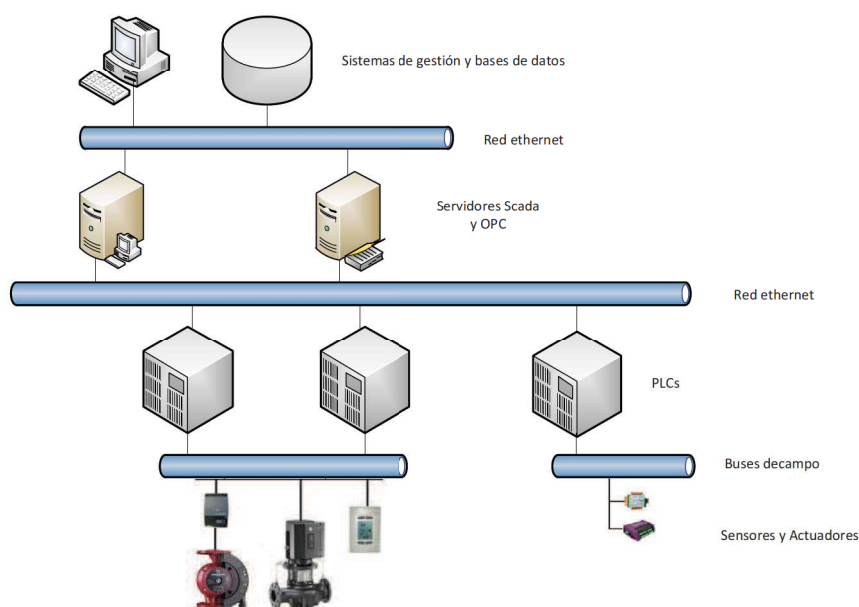


Fig. 1. Arquitectura de un Sistema de Control Automático de Procesos Industriales

rodean tanto a los sistemas HMI como a los buses de campo y a los PLCs.

Los sistemas industriales que utilizan estos tres elementos se ajustan a una arquitectura jerárquica en diferentes niveles como se muestra en la Figura 1. Así, los sistemas HMI (Scada y pantallas táctiles) se comunican con los PLCs mediante alguna red de comunicaciones, generalmente Ethernet. En cambio los PLCs se comunican y realizan acciones de control sobre los sensores y los actuadores mediante un bus de campo. La utilización de un bus de campo asegura que se verifiquen las restricciones de tiempo real necesarias para que los dispositivos reciban o envíen información al PLC y éste tome las decisiones necesarias para regular el proceso industrial para el que el sistema ha sido diseñado [9].

En este escenario nos encontramos con varias problemáticas que hacen que los sistemas industriales permanezcan hermetizados y resulte difícil y costoso la interacción con ellos. Dificultades presentes no solo con sistemas de nivel superior, como sistemas *Manufacturing Execution System* (MES) o *Enterprise Resource Planning* (ERP), sino también con sistemas utilizados en el mismo nivel pero de diferentes fabricantes, lo que los hace incompatibles entre sí. En este caso, los fabricantes desarrollan sus propios sistemas “propietarios” con sus propios PLCs, los cuales pueden utilizar distintos buses de campo y distintos sistemas HMI. Por ello, uno de los grandes retos que hay actualmente en los sistemas industriales es promover y converger a la integración, tanto de forma vertical, con sistemas de niveles superiores, como horizontal, con sistemas de otros fabricantes.

En el primer caso, en la integración con sistemas de niveles superiores, hay soluciones que dan soporte a los sistemas MES. Estas soluciones integran, por ejemplo, todo el sistema de producción de una fábrica, cubriendo aspectos como el lanzamiento de órdenes de

producción, control de inventario, ejecución de recetas, ejecución de BATCH y control de proceso en un mismo sistema. Con estas técnicas se requiere que un sistema superior indique a los sistemas de control qué acciones han de realizar y cuando han de ejecutarlas. Por tanto, ambos tienen que estar plenamente integrados y tienen que ser capaces de comunicarse para evitar errores. Estos sistemas ya existen comercialmente y están implantados en la industria [8].

La implantación de estos sistemas, si bien está resuelta y no supone ninguna novedad, suele ser costosa. Por norma general presentan bastantes dificultades, sobre todo en entornos en los cuales ya existe un sistema de control automático heredado, o en entornos con una gran heterogeneidad en cuanto a elementos (PLC, HMI) de distintos fabricantes.

En cuanto a la integración horizontal entre sistemas de distintos fabricantes encontramos aún más problemas. Cada fabricante tiene su propio sistema (HMI y PLC) y, por lo general, este no es compatible con los de otros fabricantes, por lo que la integración entre distintos productores es complicada.

Para solucionar la problemática entre la interacción de los productos de distintos fabricantes apareció el estándar *OLE for Process Control* (OPC) [10] de la fundación OPC, como una forma de llevar a cabo la comunicación entre los PLCs y los sistemas HMI, sobre todo a nivel de SCADA. La inserción de servidores OPC entre los PLCs y los sistemas HMI permite desacoplar los sistemas HMI de los PLCs a la vez que mejora la seguridad al tener un solo punto de acceso a los PLCs. La función básica de los servidores OPC es comunicarse con los PLCs utilizando un estándar bien definido, accediendo a toda la información que estos manejan, datos almacenados en memoria, y publicando dicha información para que sea accesible desde otras aplicaciones. Además, brindan la posibilidad de poder

modificar dicha información y que estas modificaciones se reflejen en la memoria del PLC [10].

El primer estándar definido por la fundación OPC, se basó en el acceso a estos servidores que mapean la memoria de un PLC mediante la utilización de *Distributed Component Object Model* (DCOM), tecnología para acceso a objetos distribuidos basada en sistemas Microsoft. En una segunda definición de este estándar la fundación OPC desarrolló la *Arquitectura Unificada OPC* (OPC UA), en la que además de por DCOM se pueden acceder a los elementos del servidor OPC mediante servicios web [11].

Es un hecho que en la actualidad todos, o casi todos, los sistemas de automatización industrial con cierta entidad utilizan uno o varios servidores OPC para realizar la comunicación HMI-PLC y conseguir así un acceso uniforme a la información que manejan los PLCs. De esta forma, cualquier sistema puede acceder a dicha información para, por ejemplo, almacenarla o para ser gestionada por sistemas superiores, como ejecutores de BATCH o sistemas MES. La aparición de estos elementos ha supuesto una revolución a la hora de interactuar e integrar los sistemas de automatización industriales [12].

Sin embargo, ¿qué ocurre con la comunicación OPC – PLC?, ¿existe un servidor OPC universal capaz de comunicarse con todos los PLCs del mercado? La respuesta es negativa. Cada fabricante proporciona su propio servidor OPC y, aunque existen intentos de crear servidores OPC capaces de interactuar con una gran variedad de PLCs o sistemas como controladoras de peso, encajonadoras, impresoras de cajas, etc., es imposible que contemplen todos los sistemas y, por lo general y salvo algunas excepciones de algunos fabricantes especializados, no funcionan tan bien como el que proporciona el fabricante del PLC. Por tanto, podemos llegar a encontrarnos con sistemas en los que tengamos un gran número de servidores OPC de distintos fabricantes. Esto conlleva problemas de mantenimiento y heterogeneidad de sistemas. Además de la redundancia de un componente de comunicaciones que, aunque esencialmente hace lo mismo en todos los casos, nos vemos obligados a utilizar de forma repetitiva debido al protocolo propietario utilizado por cada fabricante para comunicarse con su PLC o sistema empotrado.

III. TECNOLOGÍAS USADAS

Los pilares en los que se sustenta nuestra propuesta son la utilización de JavaES [6] como middleware que nos proporciona una abstracción del hardware [13][14] y DOHA como middleware basado en servicios para realizar la comunicación PLC-HMI [15]. La utilización de estas plataformas facilita el desarrollo y despliegue de aplicaciones en dispositivos empotrados de uso industrial y nos ayuda a solventar los problemas de interacción y comunicación con sistemas, tanto verticales como horizontales, gracias a la utilización de la metodología SOA implementada en forma de servicios DOHA.

Además de aportar una comunicación mediante servicios, se plantea también el desarrollo de un servidor OPC UA, haciendo que nuestra propuesta sea compatible con sistemas HMI ya existentes en el mercado. A continuación analizaremos con más detalle las características de estos dos middlewares.

A. JavaES

JavaES es un framework basado en Java que provee una capa de abstracción para el desarrollo de aplicaciones empotradas, lo que hace factible la implementación de un mismo código sobre distintos sistemas empotrados independientemente de su fabricante [14]. Para conseguir esto, JavaES se apoya en un modelo abstracto del hardware con una API basada en una serie de interfaces Java implementadas para cada tipo de dispositivo o placa empotrada. Además, incorpora un sistema de despliegue que no sólo facilita la implantación en distintos dispositivos, sino que además sugiere el hardware o la placa empotrada más adecuada capaz de ejecutar el programa. Durante la fase de despliegue se comprueba el hardware y, dependiendo de si el dispositivo dispone o no de un procesador Java, se desplegará una máquina virtual Java mínima adaptada a las necesidades tanto de la aplicación como de la plataforma empotrada. Además, JavaES proporciona una serie de proxies ya implementados que facilitan la depuración independientemente de las herramientas disponibles del empotrado, así como la conexión con sistemas empotrados subrogados en los que no es posible ejecutar el entorno de ejecución de JavaES.

Una de las características más interesantes de JavaES es que, una vez que un programa ha sido probado y depurado, éste puede seguir utilizándose independientemente de los cambios o actualizaciones que se realicen en los componentes hardware de control, siempre que sean funcionalmente compatibles. Es decir, si un programa es válido para un proceso, seguirá siendo válido aunque se cambie el empotrado que lo ejecuta. Esto es posible porque el modelo abstracto de hardware de JavaES virtualiza el hardware, y lo desacopla del hardware real [6]. Gracias al uso de las interfaces provistas en JavaES, el programa puede mantenerse intacto, aunque cambiemos el modo en cómo JavaES interacciona con el hardware, directamente o a través de un protocolo de comunicaciones. En caso de que las interfaces estén implementadas también para esas arquitecturas, el trabajo será mínimo y solo deberá cambiarse la clase final que se instancia [14]. En definitiva, JavaES puede ser una herramienta muy útil para desarrollar aplicaciones de automatización industriales, gracias a la reutilización, flexibilidad y capacidad de abstracción que ofrece con respecto al hardware. Las aplicaciones en JavaES son poco acopladas y poco dependientes del hardware subyacente. Incluso podríamos pensar en sistemas interoperables desde un punto de vista hardware con los consiguientes beneficios, tanto en facilidad de desarrollo, usabilidad y mantenimiento de nuestras aplicaciones. Por lo tanto, con la utilización de JavaES

se pueden llegar a reducir significativamente los tiempos de desarrollo y, por consiguiente, los costes de las aplicaciones en el ámbito de los sistemas industriales.

Cabe destacar que, gracias a la evolución de los sistemas empotrados, tanto en capacidad de procesamiento y almacenamiento, cada vez contamos con un número mayor de empotrados capaces de soportar JavaES sin apenas problemas de recursos. Muchos de estos empotrados están apareciendo en el mundo industrial como alternativas a los sistemas de automatización industrial tradicionales [11].

B. DOHA

En la actualidad existen muchos trabajos en los que se intenta acercar la filosofía SOA al mundo industrial [3][5][11]. Los trabajos más prometedores en este ámbito proponen implementaciones de estándares como DPWS, para introducir servicios web como parte de los elementos de un sistema de automatización industrial [4][16][17]. Por lo general, estos servicios son utilizados como puentes para acceder a la información de los buses de campo [4]. Aunque suponen un gran avance, pueden y deben de ser utilizados para tareas como la parametrización, supervisión y, por qué no, tareas de control de procesos industriales, siempre que dispongan de soporte para tiempo real.

En este sentido, DOHA proporciona un middleware para el desarrollo de servicios [15][18]. DOHA está basado en SOA, por lo que los servicios desarrollados en esta plataforma ofrecen entre otras propiedades, bajo acoplamiento, reconfiguración dinámica, flexibilidad, interoperabilidad, características relevantes para el desarrollo de aplicaciones de control y supervisión en sistema industriales.

DOHA establece una plataforma de servicios especialmente orientada a dispositivos con pocos recursos, para lo cual introduce restricciones asociadas al tamaño y al formato de los mensajes intercambiados entre dispositivos, además de un sistema ligero de descubrimiento de servicios. El cumplimiento de estas restricciones, garantizan la utilización de la plataforma con éxito sobre dichos dispositivos.

DOHA está implementado sobre dos tecnologías, JXTA y DPWS. La primera de ellas, JXTA, abreviatura de *Juxtapose*, motivado por el hecho de romper con el tradicional modelo cliente-servidor, sigue una arquitectura distribuida peer-to-peer (P2P), mientras que la segunda está basada en la utilización de servicios web. La Figura 2 muestra la relación que hay entre DOHA y las implementaciones en JXTA y DPWS. DOHA proporciona una abstracción con respecto a la tecnología de implementación. Dicha abstracción nos permite desarrollar una aplicación basada en servicios DOHA sin preocuparnos de la tecnología que los sustenta. Gracias a ello, podremos utilizar JXTA o DPWS indistintamente sin modificar nuestra aplicación DOHA. Esto nos permite escoger la tecnología que mejor se adapte al entorno donde queremos desplegar la aplicación basada en servicios. En este sentido podemos afirmar que DOHA es una plataforma interoperable

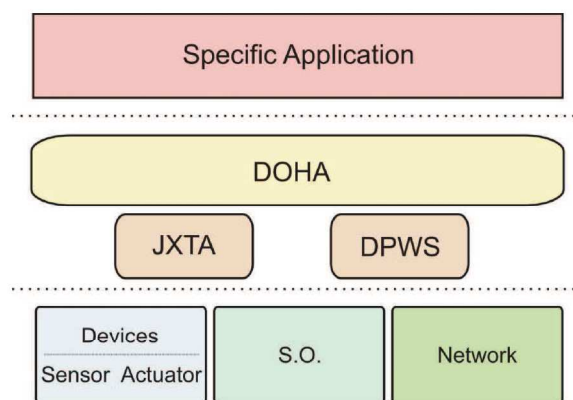


Fig. 2. Arquitectura de DOHA

desde el punto de vista de la tecnología de implementación.

Cada servicio DOHA está caracterizado por una arquitectura multicapa: interfaz, aplicación e interacción. Mediante la capa de interfaz se gestionan todas las peticiones que llegan al servicio cuando éste actúa como servicio proveedor. La capa de aplicación recoge la funcionalidad propia del servicio. Es decir, en esta capa se implementan las distintas operaciones que conforman la funcionalidad del servicio. Por último, mediante la capa de interacción se lleva a cabo la interacción con otros servicios, es decir, el servicio puede actuar como servicio consumidor con respecto a los demás servicios del sistema. Un servicio DOHA requiere además de una serie de documentos de especificación para su caracterización: un contrato de servicio, el mapa de composición y la especificación de configuración [18].

DOHA además ofrece un modelo de composición de servicios basado en el grado de complejidad de las operaciones que ofrece cada servicio [19]. Entendiendo el grado de complejidad de una operación como la necesidad de interconexión con otros servicios para completar su funcionalidad, podemos distinguir entre servicios simples, como aquellos que sólo tienen operaciones que no requieren interconectarse con ningún otro servicio, y servicios compuestos con operaciones compuestas, que requieren la invocación de otras operaciones en otros servicios. DOHA garantiza, por construcción, que no se formen ciclos al invocar operaciones de otros servicios evaluando el grado de complejidad de las operaciones compuestas de cada servicio. El modelo de composición de servicios de DOHA impone que las operaciones con un grado dado no puedan invocar a operaciones de grado superior y si a operaciones de grado inferior. De este modo se impide que operaciones de grado inferior puedan interaccionar con operaciones de grado superior, evitando así la formación de ciclos de invocaciones sucesivas de operaciones.

Las características que ofrece DOHA pueden beneficiar al desarrollo de aplicaciones en el ámbito de los sistemas de control industrial, ya que el uso de servicios puede ayudar a mejorar los sistemas de monitorización y configuración. DOHA facilita la

creación de nuevos servicios a partir de la integración de los datos que se obtienen al colaborar con otros servicios, lo que puede ayudar a los operarios en la monitorización del sistema y en su toma de decisiones. Así, es posible, por ejemplo, implementar diferentes servicios encargados de obtener valores de una magnitud como la temperatura en un entorno con diferentes grados de precisión, y luego combinarlos, bien para obtener una medida más precisa, o bien para supervisar su variación en dicho entorno. Con este sistema cada medida la daría un servicio y otros servicios compuestos se encargarían de combinar los datos obtenidos. Al ser servicios web el operario podría verlas con un ordenador, tablet o móvil. Otra de las ventajas es que aunque un sensor dejara de funcionar el servicio podría recuperarse del error y el operario podría seguir viendo la medida, lo cual aumenta la robustez del sistema. Otro posible uso sería la creación de un servicio fusor de datos que genere nueva información a partir de las distintas medidas en una cadena de producción facilitando la monitorización del conjunto e incluso detectando situaciones críticas de forma que los operarios puedan adelantarse a posibles errores. Y por último, podríamos utilizar servicios DOHA como forma de comunicación entre los sistemas HMI y los PLCs, consiguiendo desarrollar sistemas HMI, SCADAS o pantallas táctiles que se comuniquen directamente con los PLCs realizando llamadas a servicios DOHA, abriendo la posibilidad de que sistemas superiores, como los ya mencionados sistemas MES, puedan comunicarse con PLCs simplemente mediante la invocación de servicios DOHA, con las consiguientes ventajas que ello supondría, tanto en flexibilidad, costes y facilidad de integración. Además, cabría la posibilidad de desarrollar un servidor OPC basado en DOHA siguiendo el estándar OPC UA [11] y haciendo que las aplicaciones o desarrollos HMI de mercado que hoy en día existen puedan utilizar DOHA como sistemas de comunicación con los PLCs que controlan los procesos industriales [14].

IV. INTEGRACIÓN DE JAVAES Y DOHA

En secciones anteriores se han presentado dos tecnologías cuya combinación puede ayudar a mejorar el desarrollo de sistemas industriales. Para ello es necesario unirlos e integrarlas en el mismo sistema a fin de obtener un sistema completo y funcional.

JavaES fundamentalmente proporciona una capa de abstracción de hardware de señales de entradas y salidas analógicas y digitales, que puede adaptarse a diferentes configuraciones con dispositivos I/O físicos. Por ejemplo, podemos trabajar con tarjetas de I/O analógicas y digitales directamente, o bien utilizar un bus industrial para conectarnos directamente al PLC o un dispositivo I/O. Dicha versatilidad facilita que los cambios que se realicen en el hardware tanto a nivel arquitectónico como a nivel de comunicaciones no

afecten a la funcionalidad de la aplicación sobre JavaES, siempre que estos cambios sean por elementos hardware con las mismas características funcionales.

En cambio DOHA proporciona un middleware de servicios en el cual las aplicaciones se implementan en base a la colaboración de uno o varios servicios, que pueden estar ejecutándose en un mismo dispositivo o en varios. Cada uno de estos servicios puede responsabilizarse de aspectos de bajo nivel ligados a los dispositivos I/O como sensores o actuadores para la monitorización y modificación de las señales de entrada/salida, o de aspectos de mayor nivel como reguladores, planificadores, etc. En cualquiera de los casos, el uso de sistemas abiertos posibilitaría la convergencia tanto de sistemas verticales como horizontales del sistema de automatización, sin necesidad de utilizar los protocolos propietarios de los fabricantes, con las consiguientes ventajas, ya mencionadas, que esto supondría [20].

La integración de DOHA y JavaES en una misma aplicación para un sistema de automatización industrial nos permite implementar una aplicación en base a la interacción entre varios servicios DOHA. Los servicios DOHA de bajo nivel que encapsulan el hardware pueden acceder al mismo utilizando el modelo de abstracción de JavaES. Independientemente de donde se encuentre el dispositivo de I/O, JavaES se encargará de extraer o actualizar las señales en los dispositivos de I/O. Por otra parte, la colaboración de los servicios DOHA nos permite desarrollar las aplicaciones HMI o nuevos servicios u aplicaciones de más alto nivel a partir de las señales que proveen los PLCs.

No obstante, no debemos de olvidar los problemas que podemos encontrarnos a la hora de desplegar estas dos plataformas en ámbitos tan conservadores y herméticos, sin contar además con las características del hardware que se ha venido utilizando tradicionalmente en el desarrollo de sistemas industriales como la ausencia de sistema operativo, el uso de hardware y software propietario del fabricante, o la ausencia de estándares. En el caso de los estándares en los últimos años han ido apareciendo estándares tanto en buses de campo y sistemas de comunicación PC-PLC con OPC. A esto le tenemos que sumar, además, la poca penetración que ha tenido la tecnología Java en entornos industriales.

Este escenario tan poco prometedor, que en principio podría hacernos pensar que estamos abordando una tarea imposible, está sufriendo cambios; cambios que están haciendo que tecnologías como los servicios web se estén intentando introducir en el dominio de las aplicaciones industriales [20]. Si a estos desarrollos además le sumamos los cambios que se están produciendo a nivel hardware con la aparición de PLCs en dispositivos empotrados basados en sistemas operativos (Soft-PLC), se puede apreciar un cambio de tendencia que nos posibilita el despliegue de DOHA y JavaES en ambientes industriales.

V. CASO DE ESTUDIO

Para probar que la propuesta que se propone en el artículo es viable hemos desarrollado un prototipo del sistema basado en la utilización de dispositivos empotrados del fabricante *Texas Instrument* [21]. En particular hemos seleccionado las placas AM3359 ICE (Industrial Communication Engine) y AM3358, que pueden ser utilizadas para el desarrollo de aplicaciones industriales.

La plataforma empotrada AM3359 ICE tiene un procesador ARM Cortex 8, 256 Mb de RAM y soporte para tarjetas micro SD. El soporte de tarjetas micro SD nos permite arrancar los binarios del programa y del sistema operativo, y posteriormente almacenar los datos que se han generado. Lo interesante de este empotrado es que nos da la posibilidad de conectarnos como esclavo a la gran mayoría de los buses de campo industriales, PROFIBUS, PROFINET, EtherCAT, CAN, etc. La placa cumple según el fabricante con los requisitos propios de entornos industriales por lo que puede actuar como sistema de I/O para un amplio abanico de aplicaciones industriales, con o sin bus de campo. Dispone además de soporte para dispositivos I2C, GPIO, SPI y UART. También ofrece la posibilidad de poder controlar actuadores como motores eléctricos directamente, mediante drivers, proporcionados por el propio fabricante.

El segundo empotrado que vamos a utilizar es el AM3358. Este empotrado dispone de un procesador ARM Cortex 8, 256 Mb de RAM y soporte para tarjetas micro SD. En este caso el dispositivo nos permite trabajar con Linux, QNX [20] y Android. También tiene soporte para trabajar con EtherCAT, en cuanto a buses industriales, y además soporta I2C y GPIO. Aparte de este tipo de conectividad también dispone de conectividad WIFI y Bluetooth, y una pantalla táctil que nos da soporte para el desarrollo de un sistema HMI.

La utilización conjunta de ambas placas nos permite desarrollar aplicaciones industriales a través tanto de JavaES como de DOHA. Existen varias posibles disposiciones o configuraciones posibles en función de donde colocamos el dispositivo de I/O o PLC. La primera configuración o *configuración 1* es la que utiliza sistemas empotrados tanto para el acceso a las señales de entrada/salida como para la aplicación HMI. La Figura 3(a) muestra la disposición de los empotrados según su papel de esclavo o maestro en el bus industrial. En esta configuración, el empotrado AM3359 se ha utilizado como sistema de I/O, es decir, un esclavo dentro de la típica topología maestro-esclavo presente en buses de campo, y se ha conectado a un motor eléctrico. El motor se conecta directamente al dispositivo mediante un driver del propio fabricante para las I/O digitales. Por otra parte, en la placa AM3358, hemos desplegado el middleware JavaES-DOHA. JavaES nos permite aislar los detalles de la comunicación con los dispositivos de I/O conectados a través del bus industrial EtherCAT [7], y extrae/actualiza el estado del motor. Hemos implementado los servicios que se encargaran de

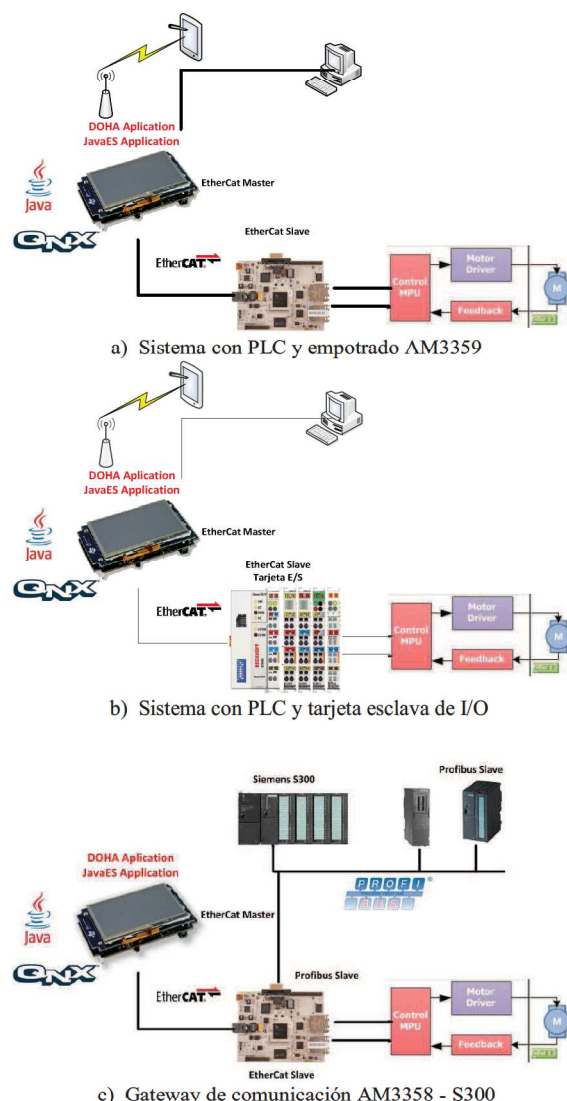


Fig. 3. Configuraciones de interconexión con sistemas industriales

monitorizar o parametrizar los variables del sistema de control que actúan sobre el motor, como la velocidad de giro, y la dirección de giro. También se ha implementado un servicio actuador de DOHA para cambiar tanto la velocidad de giro del motor como el sentido de giro a través de dos operaciones que realizan estas acciones sobre el dispositivo I/O utilizando JavaES. Por otra parte, se ha implementado otro servicio DOHA para leer el estado del motor (velocidad actual y el sentido de giro del motor) mediante la ejecución de otras dos operaciones de JavaES sobre el dispositivo a través del bus industrial.

Estos dos servicios, Figuras 4 y 5 son utilizados dentro del mismo dispositivo para dar soporte a una pequeña aplicación HMI para gestionar el sistema de control del motor. Además pueden dar soporte a otra aplicación con la misma funcionalidad HMI que se ejecuta sobre un PC o sobre un dispositivo móvil (Smartphone o Tablet) con Android, mediante la utilización de conexiones cableadas o inalámbricas.

En la segunda configuración o *configuración 2* hemos sustituido la placa AM3559 por una tarjeta de I/O (e.g. Beckhoff) que actúa de esclavo EtherCAT en lugar del dispositivo EM3359, tal y como se muestra en la Figura 3(b). En este caso no tendríamos que reprogramar la aplicación ni modificar ninguno de los servicios implementados, dado que la aplicación trabaja sobre JavaES utilizando el modelo abstracto del hardware en base a entradas y salidas virtualizadas. Por tanto, la sustitución de un dispositivo por otro se consigue estableciendo la correspondencia entre las entradas/salidas virtuales con las físicas. Evidentemente, JavaES tiene que tener soporte para el bus industrial y en determinados casos para el dispositivo de entrada/salida concreto [14]. Para ello, JavaES cuenta con un mecanismo de extensión hardware que permite la adición de nuevos dispositivos y empotrados controlados por JavaES.

La tercera configuración o *configuración 3* ha consistido en combinar dos buses de campo para resolver un problema que se presenta en algunos casos en instalaciones industriales. Es la situación que se presenta cuando, para manejar un equipo o un sistema, se necesita utilizar un PLC con sistema propietario que además trabaja con un bus de campo específico, diferente del bus de campo utilizado para el resto de PLCs de la instalación, y nos surge la necesidad de comunicar los PLCs de nuestra instalación para intercambiar información y coordinar dos procesos industriales diferentes. En la Figura 3(c) se muestra una posible disposición de la instalación, en la que se ha utilizado la placa AM3559 como puente para realizar comunicaciones PLC-PLC, EtherCAT y PROFIBUS [22].

Los servicios se siguen ejecutando en la placa AM3358, y el acceso al hardware se sigue realizando a través de JavaES sobre el dispositivo I/O que se encuentra en EtherCAT. La diferencia está en la implementación que se tiene que realizar en el empotrado que actúa de puente, es decir, en el AM3359. En este caso concreto, se ha implementado una aplicación sobre JavaES directamente en el AM3359 para abstraer los procesos de planta teniendo en cuenta las peculiaridades de cada bus de campo. La aplicación actúa de gateway o pasarela replicando las peticiones y respuestas entre los dispositivos S300 de Siemens que utiliza como bus de campo PROFIBUS, y el AM3358 que utiliza como bus de campo EtherCAT. Gracias a la abstracción que hace JavaES de EtherCat y de PROFIBUS, podemos realizar esta comunicación hacia la placa AM3358 sin preocuparnos de los detalles de los buses de campo, y poder enviar y recibir datos del Siemens S300. En el caso de que por necesidades de ampliación del sistema tuviéramos que utilizar otro bus de campo como, por ejemplo, PROFINET tan solo tendríamos que utilizar la abstracción de JavaES de ese bus de campo, y el sistema podría comunicarse con el PLC que controle este bus de campo.

En teoría, por tanto, podríamos portarlo a diferentes tipos de buses de campo como PROFIBUS, EtherCAT,

Operaciones y Servicios DOHA Implementados

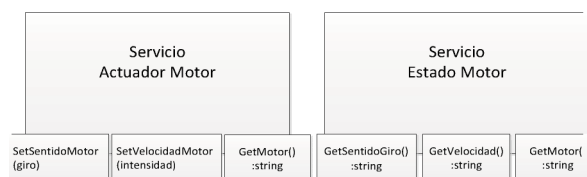


Fig. 4. Servicios DOHA Implementados

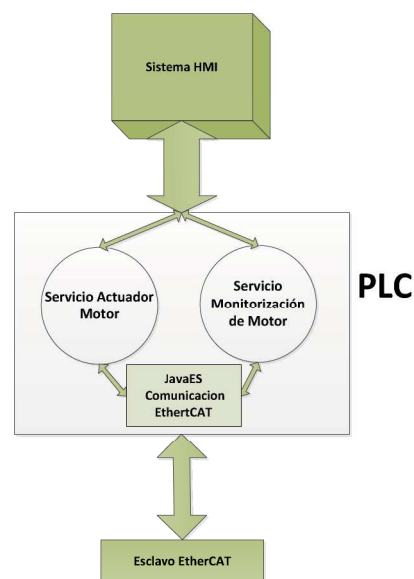


Fig. 5. Esquema de los servicios ejecutados en AM3358

PROFINET, CAN, etc., y conseguir realizar la comunicación PLC-PLC a través de buses de campo distintos. Sin embargo, hemos tenido algunos problemas con el bus de campo PROFIBUS, y la aplicación gateway sobre JavaES no funciona correctamente.

Está claro que este esquema ofrece grandes ventajas a la hora de desarrollar sistemas verticales, gracias a la utilización de la filosofía SOA para interactuar con los PLCs, pero también obtenemos grandes ventajas en la interacción/comunicación con sistemas horizontales en el mismo entorno industrial gracias a JavaES como middleware de interacción con el hardware.

VI. DISCUSIÓN Y TRABAJOS FUTUROS

Puesto que este artículo se basa en una investigación en progreso, todavía no hemos podido realizar pruebas detalladas de rendimiento ni estudiar la sobrecarga que puede tener la utilización de DOHA y JavaES en el desarrollo de aplicaciones para sistemas industriales. Los primeros resultados obtenidos muestran que propiedades como la flexibilidad, la reutilización y la extensión del sistema son mucho mejores que los que se obtienen con los sistemas propietarios del mercado.

Se espera obtener nuevos resultados y técnicas que se basen en la combinación de este tipo de tecnologías para

ofrecer un mayor acceso a los sistemas industriales que permitan la creación de sistemas de control y monitorización de forma más sencilla y menos rígida que en la actualidad.

En cuanto a JavaES, hay que mejorar el soporte de buses industriales y de dispositivos de I/O, siempre que tengamos soporte hardware para ello. En sistemas propietarios podría haber dificultades debido a la falta de documentación o acceso al hardware. En estos casos podríamos usar arquitecturas subrogadas en las que algún tipo de empotrado pudiese conectarse a estos buses e interactuar con ellos.

Se ha probado con éxito la interacción con un empotrado y un PLC sólo con EtherCAT de acuerdo a la con las conexiones que se muestran en la figura 3. Esperamos que pueda extenderse a otros buses de campo como PROFIBUS, PROFINET y CAN, pero necesitamos profundizar en la diferencias que hay entre los diferentes tipos de buses de campo. Se nos plantean desafíos como la interacción de servicios DOHA capaces de enviar y leer información directamente del PLC de control de proceso, o por qué no, a través de un bus, estudiar la eficiencia de los mismos con respecto a los sistemas industriales actuales y, plantearnos la posibilidad de realizar control industrial directamente con servicios DOHA.

Por otro lado, configuraciones en instalaciones industriales como las que se muestran en la figura 3 configuración 3 nos posibilitaría plantear la coordinación de varios PLCs para controlar ciertos procesos industriales. Todavía tenemos que realizar más pruebas para ver si el gateway entre buses de campo es factible, y en consecuencia, podemos extender el uso sobre varios tipos de buses.

Los servicios DOHA implementados sobre el maestro pueden proveer de toda la información manejada por los PLCs para la supervisión y monitorización de la planta, incluso garantizando restricciones de tiempo real. Sin embargo, debemos profundizar en las consecuencias que esto tiene en niveles superiores, y los límites que tendríamos respecto al tiempo real.

Otra aspecto importante, y que requiere mucho más trabajo, es la posibilidad de dar soporte a sistemas que ya estén implantados, sistemas HMI (SCADA fundamentalmente), para así conseguir una compatibilidad con todos los sistemas existentes en el mercado. Se plantea el desarrollo de un servidor OPC UA, utilizando servicios DOHA, para la parte de comunicación con el PLC, por lo que tendríamos un componente de comunicación con nuestros PLCs de proceso compatible con los sistemas existentes y, por consiguiente, con todas las soluciones implantadas hoy por hoy en el entorno de las aplicaciones industriales.

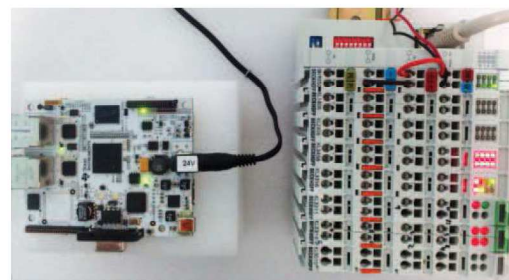


Fig. 6. Conexión entre AM3359 y tarjeta de E/S mediante EtherCAT

La utilización de estos middleware para el desarrollo de sistemas de control industrial ofrece muchas posibilidades que deben de ser exploradas con el fin de aportar soluciones al desarrollo de nuevos sistemas para el control automático en entornos industriales capaces de solventar los problemas existentes, mejorar el desarrollo de este tipo de aplicaciones e introducir conceptos nuevos como servicios, interoperabilidad, flexibilidad y reusabilidad en el entorno de los sistemas de control industrial. Conceptos que hasta hoy son casi exclusivamente de otros dominios, como las aplicaciones de negocio.

VII. CONCLUSIONES

En este artículo se han presentado dos middleware, DOHA y JavaES, para el desarrollo de aplicaciones industriales, que aportan ventajas como son la capacidad de reutilización, la interoperabilidad, la facilidad de uso, y la capacidad de abstracción del hardware subyacente. Por un lado, esto nos permite disminuir los tiempos de desarrollo de nuestras aplicaciones y aislarlas de las características del hardware o del bus de campo utilizando. Por otro lado, permite mejorar la integración/interacción con otros sistemas superiores como sistemas MES y ERPs.

Con JavaES conseguimos ganar en reutilización y disminuir los tiempos de prueba para aplicaciones que migran de un hardware a otro, consiguiendo, como ya se ha mencionado, una abstracción del mismo que nos garantiza que, si una aplicación funciona sobre un hardware, al cambiarlo por otro del mismo tipo compatible, la aplicación va a seguir funcionando, aunque el hardware sea de otro fabricante. Esta capa de abstracción nos permite ofrecer estos datos a otros sistemas de una forma bastante sencilla, por lo que los problemas de integración entre sistemas horizontales pueden llegar a mitigarse bastante; por ejemplo, mediante proxies que actúan de gateway entre distintos tipos de buses de campo.

Al combinar esta tecnología con DOHA conseguimos un sistema flexible e interoperable, que puede ser utilizado por cualquier sistema o hardware capaz de trabajar con servicios DOHA. El uso de servicios también permite crear sistemas más complejos con capacidad para procesar los datos [23], y ofrecer arquitecturas y aplicaciones fácilmente integrables con otros sistemas, tanto horizontalmente como

verticalmente, para el control y la monitorización de los sistemas industriales.

La utilización de estas dos tecnologías nos permite desarrollar sistemas no propietarios con tecnologías abiertas como Java en sistemas industriales. Además los middleware que estamos utilizando emplean siempre estándares como forma de trabajo implícita. En el caso de JavaES estándares de comunicación a nivel de bus y en el caso de DOHA estándares como DPWS basados en SOA. Por lo tanto, cualquier sistema compatible con estos estándares también sería, en cierto modo, compatible con las aplicaciones desarrolladas con los middleware DOHA y JavaES. La combinación de las tecnologías presentadas en este artículo puede mejorar los sistemas industriales ofreciendo mecanismos útiles, flexibles, reutilizables e interoperables, tanto para la monitorización como para la parametrización y el control de los procesos industriales.

REFERENCIAS

- [1] M. Stopper and B. Katalinic, "Service-oriented architecture design aspects of opc ua for industrial applications," in *International MultiConference of engineers and computer scientifics*, vol. 2, 2009.
- [2] "Socrates: <http://www.socrates.eu/>."
- [3] G. Candido, F. Jammes, J. de Oliveira, and A. Colombo, "Soa at device level in the industrial domain: Assessment of opc ua and dpws specifications," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, pp. 598–603, 2010.
- [4] M. Izaguirre, A. Lobov, and J. Lastra, "Opc-ua and dpws interoperability for factory floor monitoring using complex event processing," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, pp. 205–211, 2011.
- [5] P. Jiang, X. Shao, H. Qiu, L. Gao, and P. Li, "A web services and process-view combined approach for process management of collaborative product development," *Computer in Industry*, vol. 60, pp. 416–427, Aug. 2009.
- [6] J. A. Holgado-Terriza and J. Viúdez-Aivar, "A flexible java framework for embedded systems," in *ACM International Conference Proceeding Series*, pp. 21–30, 2009.
- [7] E. T. Group, *Technical Introduction and Overview*, 2005.
- [8] V. Vyatkin, "Software engineering in industrial automation: State of the art review," *Industrial Informatics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.
- [9] B. Rahmani and A. Markazi, "Networked control of industrial automation systems—a new predictive method," *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 5-8, pp. 803–815, 2012.
- [10] L. Zheng and H. Nakagawa, "Opc (ole for process control) specification and its developments," in *SICE 2002. Proceedings of the 41st SICE Annual Conference*, vol. 2, pp. 917–920, 2002.
- [11] I. Seilonen, A. Tuomi, J. Olli, and K. Koskinen, "Service-oriented application integration for condition-based maintenance with opc unified architecture," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, pp. 45–50, 2011.
- [12] Q. Deng, J. Zhang, X. Xue, and Y. Zou, "Research of mes integration framework in process industry based on mas and opc technology," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2, pp. 7171–7175, 2006.
- [13] J. A. Holgado-Terriza, S. Rodríguez-Valenzuela, and J. L. Muros-Cobos, "Java para el desarrollo de sistemas empotrados," in *Actas de las II Jornadas de Computación Empotrada (JCE)*, 2011.
- [14] J. Holgado-Terriza and J. Viúdez-Aivar, "Javaes, a flexible java framework for embedded systems," in *Distributed, Embedded and Real-time Java Systems (M. T. Higuera-Toledano and A. J. Wellings, eds.)*, pp. 323–355, Springer US, 2012.
- [15] S. Rodríguez-Valenzuela and J. A. Holgado-Terriza, "A home-automation platform towards ubiquitous spaces based on a decentralized p2p architecture," in *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*, pp. 304–308, Springer Berlin / Heidelberg, 2009.
- [16] B. Bony, M. Harnischfeger, and F. Jammes, "Convergence of opc ua and dpws with a cross-domain data model," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, pp. 187–192, 2011.
- [17] G. Candido, F. Jammes, J. Barata, and A. Colombo, "Generic management services for dpws-enabled devices," in *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, pp. 3931–3936, 2009.
- [18] S. Rodríguez-Valenzuela, J. M. Gutiérrez-Guerrero, and J. A. Holgado-Terriza, "A service platform towards the interoperability between jxta and web services," tech. rep., Universidad de Granada, 2013.
- [19] J. Holgado-Terriza and S. Rodríguez-Valenzuela, "Services composition model for home-automation peer-to-peer pervasive computing," in *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, pp. 529–536, 2011.
- [20] J.-Y. Kim, Y. J. Lee, S.-W. Cheon, J.-S. Lee, and K. C. Kwon, "A commercial-off-the-shelf(cots) dedication of a qnx real time operating system (rtos)," in *Reliability, Safety and Hazard (ICRESH), 2010 2nd International Conference on*, pp. 123–126, 2010.
- [21] "Texas instruments: <http://www.ti.com/>."
- [22] P. International, *Profibus Specification*, 1998.
- [23] S. Rodríguez-Valenzuela, J. Holgado-Terriza, J. L. Muros-Cobos, and J. M. Gutiérrez-Guerrero, "Data fusion mechanism based on a service composition model for the internet of things," in *Actas de las III Jornadas de Computación Empotrada (JCE)*, vol. 1, pp. 64–69, Servicio de Publicaciones. Universidad Miguel Hernández, Elche, 2012, Elche (Alicante) 19-21 Septiembre 2012 2012.