

# SOA en Automatización de Pymes Manufactureras

Nora S. Blet<sup>1</sup>, José L. Simón<sup>1</sup>

<sup>1</sup> Dpto. de Sistemas e Informática, Universidad Nacional de Rosario (UNR)  
{nblet, jlsimon}@fceia.unr.edu.ar

**Resumen.** En la actualidad los sistemas productivos de una Pyme manufacturera no están listos para las arquitecturas orientadas a servicios (SOA) y, como consecuencia, el proceso de migración hacia la automatización basada en SOA debe realizarse paso a paso organizando la coexistencia de nuevas tecnologías con aquéllas preexistentes. De esta forma se minimiza el riesgo financiero, hecho particularmente importante en el caso de una Pyme a la vez que, se aprovecha la oportunidad para adquirir experiencia acerca de las fortalezas y debilidades del paradigma SOA. Una parte importante del proceso de migración es la integración, directa o indirecta, en el mundo SOA de los sistemas preexistentes y, objetivo de esta contribución que intenta ofrecer un panorama general del tema, primer paso en el mapa de ruta de SOA en la automatización de una Pyme manufacturera.

**Palabras Clave:** SOA, CIM, Integración, Automatización Industrial, DPWS.

## 1 Introducción

Las empresas manufactureras del siglo XXI están inmersas en un entorno altamente cambiante, con mucha presión en cuanto a tiempos de salida al mercado, donde constantemente emergen nuevas tecnologías y se incrementa cada vez más la competencia global [1, 2]. Por tanto, las estrategias manufactureras aplicadas deberían poner énfasis en soportar esta competitividad global, la fabricación de nuevos productos personalizados e innovadores, poseer alta sensibilidad a los mercados, sin olvidar factores vitales como costo y calidad. Para lograrlo necesitarán satisfacer ciertos requisitos fundamentales:

- Integrar totalmente los sistemas de software y hardware de la empresa; sea real, virtual o una cadena de suministro.
- Usar sistemas de arquitecturas abiertas que permitan acomodar nuevos subsistemas (software o hardware) y/o desmontar aquéllos existentes, todo “sobre la marcha”.
- Lograr una comunicación y cooperación efectiva y eficiente entre empresas, como así también entre departamentos de las mismas.
- Incorporar factores humanos.

- Dar rápida respuesta a cambios en órdenes externas y perturbaciones inesperadas, tanto internas como externas a la empresa.
- Proveer tolerancia a fallos tanto a nivel sistemas como subsistemas que, posibilite la detección y recuperación de los mismos, además de minimizar el impacto que pudiesen producir en el entorno de trabajo.
- Promover agilidad, reconfigurabilidad y escalabilidad.

Los sistemas de automatización industriales tradicionales están contruidos generalmente de forma rígida, centralizada, jerárquica, propietaria y, basados principalmente en controladores lógicos programables (PLC) aunque, en contraposición se distinguen por su heterogeneidad de dispositivos, redes, protocolos y aplicaciones. Todas estas características contribuyen a hacer más costosa su reconfiguración, tanto en tiempo como en dinero, para adaptarse a los nuevos requerimientos de las empresas manufactureras.

Hoy día se reconoce que dichos requisitos se satisfacen de forma razonable mediante la adopción de arquitecturas orientadas a servicios (o SOA, por *Service Oriented Architectures*) en todos los niveles de la empresa manufacturera que, además permite obtener una infraestructura de comunicación homogénea basada en un único paradigma. Sin embargo, debido a que, el software y hardware utilizados a nivel de piso de planta son significativamente diferentes de aquéllos usados en las aplicaciones empresariales y que, los ingenieros de automatización no están familiarizados con esta nueva filosofía, la adopción integral de esta orientación a servicios en la industria manufacturera constituye todo un desafío.

La razón de la elección del tema del artículo: la situación en nuestro país. Yoguel *et al.* [3] realizan en el 2002, una encuesta a 246 empresas industriales argentinas predominantemente Pymes (69%). Los resultados de la encuesta indican en cuanto a utilización de software que, la mayoría de las firmas tiene una demanda poco sofisticada, principalmente de sistemas de automatización de oficinas y estandarizados (78%). En el área de fabricación, por el contrario, la presencia de equipamiento informático y la utilización de software específico es considerablemente menor que en el área de gestión. El equipamiento más difundido es el PLC, presente en el 23% de los casos; le siguen el control numérico computarizado (CNC), los robots, el control numérico, el sistema automático de ensamblado y montaje y, la célula flexible de fabricación. En la mayoría de los casos se trata de equipos que operan aisladamente sin integrarse mediante software con otras áreas de la empresa. Sólo en diseño se observa una importante difusión de sistemas (como el de diseño asistido por computador y otros que están presentes en el 50% de los casos), mientras es menor en las actividades de planificación y control y, está ausente casi del todo en la fase de fabricación: sólo el 4% de las firmas utiliza algún sistema de manufactura asistida por computador.

## 2 De la era CIM a la SOA

La crisis del petróleo de 1973 llevó a la economía mundial a un período de bajo crecimiento. Como consecuencia de la reducción de los mercados se produce una excesiva competencia entre los fabricantes, forzados a producir bajo pedido, con plazos de entrega ajustados y a precios reducidos. Además, a fin de satisfacer la diversificación de la demanda de los mercados, debieron fabricarse una gran variedad de productos en términos de especificaciones, modelos y dimensiones.

Desde principios de los 80 y, a pesar de un cambio en las condiciones socioeconómicas de bajas a estables, la competencia entre los fabricantes se vuelve más feroz. Para sobrevivir a esta situación debieron esforzarse por desarrollar nuevos productos y mejorar la eficiencia de sus organizaciones, tratando de ahorrar energía, reducir costos y excedentes de stock, mejorar la calidad y promover la producción oportuna. Nos encontramos en un escenario donde la fabricación progresó y dejó de ser un arte para transformarse en ciencia; fue en este contexto que hacen su aparición los sistemas de manufactura integrada (o CIM por, *Computer Integrated Manufacturing*), la primera oleada de integración de las TIC en las fábricas y, consigna de una aparente revolución en la gestión de procesos de negocios industriales. Se tenía grandes expectativas en que plantas fabriles completamente automatizadas, acorde al estado del arte de las TIC, resolverían los problemas de costo y calidad. La visión del futuro que se tenía por esos tiempos era la de una fábrica desierta, sin trabajadores; la realidad tenía una apariencia bastante diferente de la imaginada.

El término CIM fue propuesto inicialmente por el Dr. Joseph Harrington. A pesar que Harrington no quiso acuñar un nuevo acrónimo, a principios de los 80 CIM se vuelve uno usado comúnmente tanto en la industria como en las universidades, con distintos significados dependiendo de su aplicación. Aunque, en su descripción inicial, Harrington consideró a CIM como una estructura de control y comunicación para integrar los sistemas de producción de la época, supercompartimentados y sobre especializados; en realidad CIM es un proceso de uso de computadoras y redes de comunicación para transformar islas de, las en su momento llamadas tecnologías avanzadas de fabricación basadas en computadoras, en un sistema de producción altamente integrado. El principal elemento en este concepto de integración era una base de datos común, mediante la cual compartían información las distintas unidades o grupos de usuarios de la empresa fabril.

A pesar de su potencial y de algunos casos reales victoriosos, la implementación exitosa de CIM se dificultó debido a un amplio rango de cuestiones técnicas y humanas. El problema técnico más crítico fue el desarrollo de una interfaz factible que permitiese la comunicación eficiente entre estas islas de automatización. Para intentar subsanarlo se realizaron considerables esfuerzos en el desarrollo de protocolos de comunicación viables y estándares internacionales relacionados con CIM, a pesar que cuando el término CIM se acuñó no existía la gran variedad de redes de comunicación que se tiene hoy día. Al comienzo de los 90 la crisis CIM era evidente, las tecnologías necesarias aún no estaban maduras y los trabajadores fueron sobreexigidos.

Para satisfacer los nuevos requerimientos de las industrias manufactureras y a fin de subsanar las dificultades de CIM, se propusieron distintos paradigmas o filosofías, derivados o complementarios: sistemas multiagentes, gestión de la cadena de suministros, empresas virtuales, sistemas de fabricación holónicos e integración de aplicaciones basadas en Web, etc. Mientras que, estas soluciones enriquecieron las tecnologías de gestión de la industria manufacturera y paliaron en diferentes formas la incompetencia de CIM, se solapaban unas con otras enfocando la tarea de gestión desde distintos puntos de vista determinados por sus diferentes orígenes y propósitos.

Otros paradigmas que emergieron: manufactura sin desperdicios, fabricación ágil, fabricación flexible, sistemas de fabricación reconfigurables, fabricación justo a tiempo, ingeniería concurrente, personalización masiva, sistemas de gestión de la calidad total, fabricación bajo pedido. Resulta muy difícil dictaminar si alguno de estos enfoques es mejor que los otros, sin considerar las condiciones particulares de una empresa específica. Uno de los principales obstáculos para la implementación de estos métodos es la postura conservadora usual en la industria que, al adoptar una filosofía de probada eficacia durante años es renuente embarcarse en un nuevo paradigma [4]. Otro gran impedimento es la integración con sistemas preexistentes (*legacy systems*).

Los sistemas de automatización industriales se representan generalmente por la llamada pirámide CIM o de automatización para visualizar la estructura jerárquica que es la más común en este tipo de sistemas. La introducción en la automatización de tecnologías basadas en Internet provocó que la pirámide, que en un principio tenía hasta 6 niveles, colapsara a un máximo de 3 sin importar si el modelo de la planta involucra más [5]. La adopción de una jerarquía de 3 niveles: Sistemas de gestión (o ERP, por *Enterprise Resource Planning*), MES (por *Manufacturing Execution Systems*) y piso de planta (o *shop floor*), representa la situación típica actual desde el punto de vista del software de aplicación.

En el 2000 se establecen las bases para un renovado intento hacia la integración. La ISA (*International Society of Automation*) concibe una serie de estándares para empresas manufactureras y la integración de sistemas MES/ERP bajo el nombre ISA-95, definido mediante terminología y numerosos modelos orientados a objetos. Adoptado a nivel internacional como IEC 62264 y, actualizado en 2001 y 2006. La publicación de este estándar coincidió con la adopción por parte de la industria, de la tecnología informática de XML (por *eXtensible Markup Language*) y *Web Services* (WS), para el intercambio de datos de negocios dentro de las empresas o entre ellas. La tecnología WS constituye el vehículo de implementación preferido para las SOA. El estándar ISA-95 deja muchos grados de libertad para su implementación, como contrapartida esta ambigüedad desalentó su adopción [6]; es por esto que el *World Batch Forum* crea B2MML (por *Business to Manufacturing Markup language*), implementación XML del estándar ISA-95 para importar datos relativos a pedidos y productos desde sistemas ERP y, compatible con WS.

Actualmente los sistemas MES, intentan salvar la brecha entre piso de planta y los sistemas de gestión que se ejecutan a nivel empresarial “detrás de la escena fabril”. Debido a esta brecha, la comunicación entre estos dos niveles en la mayoría de los casos, se realiza en forma manual mediante transferencia de archivos o usando protoco-

los propietarios para tipos específicos de dispositivos en piso de planta [7]. En la práctica, la solución de automatización adoptada determina el software MES utilizado; de este modo la integración resulta inflexible, propensa a errores e inoportuna.

Por razones de interoperabilidad, las actuales soluciones ERP se basan en WS que, permiten además la integración horizontal con proveedores y clientes, a la vez que posibilitan una adaptación flexible a las nuevas condiciones del mercado [8]. Desde el punto de vista de las aplicaciones la tendencia es a reducir aún más la jerarquía de automatización de 3 capas. La aplicación de conceptos modulares en la automatización de plantas fabriles permite que, las nuevas herramientas ERP y MES, puedan de esta forma aplicarse a empresas más pequeñas. Este hecho lo apoya la utilización, en el campo del control, de arquitecturas PC estándar (en cuanto a recursos computacionales e interfaces de redes), como así también el uso de software ERP con módulos funcionales que hasta ahora pertenecían a MES. Sin embargo, este enfoque MES principalmente distribuido, impondrá requerimientos más demandantes para soportar una mayor flexibilidad. El manejo de conceptos del mundo de las TIC, tales como el de las ontologías para describir formalmente información semántica relacionada con los procesos, introducirá más niveles de abstracción.

Contrariamente al principio de la era CIM, hoy día existen componentes tecnológicos básicos muy potentes pero, aún es evidente la necesidad de capas abstractas de interoperabilidad. El problema principal no ha cambiado mucho a través de los años: encontrar interfaces apropiadas entre un montón de aplicaciones que en su mayoría son no interoperables. La adopción de WS como columna vertebral de comunicación en la empresa, permitiría eliminar el altamente costoso software MES; hecho principalmente importante para las Pymes debido al ahorro que podría obtenerse [8].

Integrando WS embebidos en dispositivos del piso de planta (“*smart objects*”), se permitiría que éstos interactúen a la perfección con aquellos sistemas que se ejecutan “detrás de escena”. En este contexto la expresión “*smart objects*” se usa para dispositivos habilitados para WS, tales como, redes de sensores y dispositivos RFID. Su integración con aplicaciones a nivel empresarial constituye todo un desafío; los servicios a nivel de dispositivos presentan mayor granularidad, exhiben menor fiabilidad especialmente si se conectan en forma inalámbrica, presentan mas dinamicidad y están enfocados en aspectos técnicos antes que en aquéllos de negocios. También complica la implementación los limitados recursos computacionales y, reducidas capacidades de comunicación y almacenamiento [9]. La idea es encapsular la heterogeneidad y, mediante este paradigma, proveer interfaces estandarizadas de servicios; se intenta recrear en una empresa manufacturera la llamada “Internet de las cosas”. Sin embargo, debería encontrarse una solución que no asuma solamente cierta “inteligencia” en los dispositivos de campo y sistemas de comunicaciones basados en IP, sino que, tenga en cuenta además los sistemas preexistentes sin necesidad de cambiarlos, usando *gateways* o *proxys* que “envuelvan o adapten” el acceso a los mismos y realicen la integración real [10].

Tampoco hay que olvidar los requerimientos de tiempo real a nivel de piso de planta que, en la práctica complican la aplicación del enfoque SOA. Groba *et al.* [10] consideran que, hoy día y teniendo en cuenta estas exigencias, SOA no es aplicable a

todos los campos de la industria manufacturera; por tanto aún no puede considerarse una solución holística. Si bien, estas arquitecturas aún no satisfacen requerimientos de tiempo real y predictibilidad, es de esperar que en un futuro cercano lo hagan, tal como ocurrió con Ethernet en aplicaciones industriales. Los autores proponen identificar áreas apropiadas para su aplicación, donde no se expongan requerimientos de tiempo real, tal como es el caso de la gestión del mantenimiento, u otras sin exigencias de tiempo real críticas. Otros dominios donde los WS aún no están maduros para su aplicación son aquellos donde se exige un control de posicionamiento preciso, tal como es el caso de los sistemas de robótica, máquinas con CNC y para medición por coordenadas (CMM).

### **3 SOA en sistemas de manufactura**

#### **3.1 Conceptos básicos de SOA y WS**

Jammes y Smit [1, 4] proveen una definición que, aunque incompleta, incluye dos de las características más importantes del paradigma SOA, que parecen estar en contradicción: “Una arquitectura orientada a servicios es un conjunto de principios de arquitectura para construir sistemas autónomos e interoperables”. Las principales particularidades presentes en casi todas las definiciones de SOA son:

- Autonomía: todos los servicios son independientes y estructuralmente desacoplados.
- Interoperabilidad: alcanzada especificando los servicios ofrecidos y patrones de interacción.
- Independencia de la plataforma: Los servicios se describen idealmente mediante formatos basados en texto, tales como XML, WSDL (por *Web Services Description Language*), etc. Estas representaciones son independientes de cualquier tecnología, lenguajes de programación, arquitecturas y plataformas.
- Encapsulamiento: Los servicios ofrecen sus funcionalidades a través de interfaces bien definidas ocultando cualquier complejidad
- Facilidades para el descubrimiento de servicios: Los servicios pueden publicarse y quedar disponibles para su uso.

Una arquitectura orientada a servicios consiste esencialmente en una colección de servicios junto con infraestructura para la comunicación entre ellos, la cual puede ser tan simple como el paso de mensajes o, tan compleja como en el caso de dos o más servicios coordinados para llevar a cabo alguna actividad en una capa más alta. El concepto se conoce desde las épocas de CORBA, la primera infraestructura que integra aplicaciones ejecutándose en plataformas heterogéneas. Sin embargo, SOA no es sólo una arquitectura sino también, un modelo de programación que define una nueva forma de pensar acerca de la construcción de sistemas software [11].

Un servicio es una función bien definida, autocontenida (es decir, siempre proporciona la misma funcionalidad independientemente de los otros servicios) y, que no depende del contexto o estado de otros servicios. Los servicios pueden combinarse

dinámicamente en tiempo de ejecución, para constituir aplicaciones compuestas (como si fuesen muñecas rusas) ya sea, ensamblando aquéllos que se ejecutan encima de aplicaciones preexistentes u otros creados como parte de nuevas aplicaciones. Este proceso de composición tiene dos vistas complementarias: orquestación y coreografía; la distinción entre ellas es la misma que entre los conceptos de cooperación/coordinación y comunicación, respectivamente. Se trata de extrapolar la idea de  *mashup*  usada para combinar aplicaciones en la Web 2.0.

La tecnología WS constituye el vehículo de implementación preferido para las SOA. Para el grupo de trabajo del *World Wide Web W3C* la definición de WS es: "... un sistema software diseñado para permitir la interoperabilidad entre máquinas conectadas a través de una red, con una interfaz descrita en un formato procesable por una máquina (específicamente WSDL) y que, puede interactuar con otros sistemas de la forma prescrita en su descripción, mediante la utilización de mensajes SOAP (por *Simple Object Access Protocol*) típicamente a través del protocolo HTTP, mediante una serialización XML en conjunto con otros estándares Web".

### **3.2 Las fallas de CIM fueron oportunidades para SOA**

Mientras que existen diversos estudios y testimonios elogiando CIM, resulta más difícil encontrar ejemplos donde se reconozcan sus fallas. Ocurre que, hay una tendencia natural a considerar a la tecnología como panacea y enfatizar algunos ejemplos "cumbres" en la implementación de CIM, tales como los muy conocidos de la planta de IBM en Japón, el ejemplo de General Electric, etc.

Según Scheer *et al.* (2007) [12] faltaron algunas precondiciones necesarias para la implementación exitosa de CIM, entre otras: el uso de bases de datos como elemento de integración no se generalizó en toda la empresa y, el desarrollo de la interconexión mediante redes no era suficiente para las necesidades de integración de aplicaciones en la industria manufacturera. Asimismo señalan que la principal idea de CIM se centraba en el esfuerzo para definir y desarrollar sistemas de software grandes y monolíticos, que pudiesen manejar la integración de los distintos puntos de procesos internos de manufactura y, su principal debilidad, la incapacidad para lograrlo. Generalmente se obtenían soluciones a medida que no podían reutilizarse.

En los sistemas basados en SOA la "inteligencia" se distribuye en una gran población de pequeños dispositivos embebidos, con alto nivel de granularidad, conectados mediante redes de comunicación, débilmente acoplados e interactuando con el entorno físico y organizativo. La reprogramación de los grandes y rígidos sistemas monolíticos se reemplaza por la reconfiguración de unidades embebidas.

Otros autores opinan que la razón del fracaso de CIM residió en que los sistemas de integración resultaron extremadamente complejos de planificar, construir, operar, mantener o reconfigurar y de implementación costosa tanto en tiempo como en dinero. Zuehlke [13] aconseja recordar las lecciones aprendidas de la crisis CIM: reducir la complejidad por estricta modularización y el uso de tecnologías "sin desperdicios".

Según Blecker *et al.* [14] la falla de los primeros enfoques CIM demasiado centralizados (hardware/software) demostró que es necesario utilizar una arquitectura

de información al menos parcialmente descentralizada. La aparición de tecnologías basadas en Internet conduce, para subsanar este problema, a los llamados Sistemas de Integración de Aplicaciones Empresariales (o EAI, por *Enterprise Application Integration*) que trataron de interconectar los heterogéneos sistemas de información de la empresa fabril; heterogeneidad en lo referente a interfaces y tecnologías de redes como así también en cuanto a tiempo de vida del equipamiento. La desventaja de los EAI residió en que, al igual que CIM, resultaron enfoques centralizados y focalizados en la tecnología; en el 2003 se reportó que el 70% de los proyectos EAI fallaron. En este tipo de integración conocida también como “*hub and spoke*”, un sistema central realizaba las transformaciones necesarias para el intercambio de datos y conexión a todos los otros sistemas de la empresa. En la práctica la adaptación a este punto central resultó muy costosa y provocó muchos errores en la integración.

En Emerson *et al.* (2007) [6] se explica que, muchas empresas intentaron aplicar CIM y aunque fueron exitosas, la utilización de software de integración punto a punto a medida resultó costosa con respecto a la tasa de retorno obtenida, difícil de implementar y mantener; la evidencia de tal afirmación fue la falta de adopción de CIM desde mediados a fines de los 90. Las empresas manufactureras reconocieron que este tipo de interfaces no eran una solución adecuada; existía una pronunciada necesidad de obtener interoperabilidad entre sistemas, basada en estándares internacionales. Aunque EAI representó una mejora con respecto a la poco escalable integración punto a punto propuesta por CIM, dejó huecos que llenaron las SOA. En Groba *et al.* [10] se utiliza una implementación SOA llamada *Enterprise Service Bus* (ESB), donde una arquitectura tipo *bus* permite conectar directamente cada servicio o, indirectamente si es una aplicación obsoleta, evitando el gran número de conexiones punto a punto entre ellos. Los autores afirman que este tipo de arquitectura provee un entorno de integración altamente distribuido, levemente acoplado y basado en estándares, a diferencia de los EAI.

La mayor parte de los sistemas CIM son orientados a la tecnología; en 1981 la CASA/SME (*Computer and Automated Systems Association of the Society of Manufacturing Engineers*) publica su primer modelo conceptual de empresa manufacturera: la rueda CASA CIM que proponía un marco de integración de las islas de automatización presentes en toda la empresa, usando una arquitectura que posibilitaba compartir datos comunes entre las distintas áreas funcionales. Sin embargo estos sistemas seguían teniendo islas desconectadas del resto de la empresa manufacturera. En Blecker [14], 2005, se corrobora que 14 años después existe aún una gran cantidad de sistemas de información en islas con sólo algún grado de interconexión. Se considera que, en una típica empresa manufacturera de gran tamaño, existen entre 30 y 50 aplicaciones no integradas con el resto.

La mayor parte de los sistemas CIM, organizados en forma de rígidas jerarquías con estructuras centralizadas en una computadora principal, resultan inflexibles para el azaroso entorno manufacturero. La complejidad estructural de este tipo de arquitectura de control jerárquica, crece rápidamente con el tamaño del sistema y diversidad en la producción. Además pueden paralizarse por un único punto de falla, resultan proclives a la fragilidad en la planificación, presentan sobrecargas en los tiempos de respuesta, son muy sensibles a las perturbaciones, costosos de desarrollar y difíciles

de mantener. Zuehlke [13] en el 2010 recomienda evitar jerarquías centralizadas y favorecer el uso de estructuras descentralizadas con enlaces poco rígidos, constituidos por módulos autoadaptativos, tal como propician las potentes SOA. El reto más grande de este cambio se le plantearía a los planificadores fabriles, experimentados a través de décadas en el pensamiento jerárquico, los cuales deberían aprender nuevas estrategias que permitan componer sistemas técnicos complejos usando modelos semánticos de servicios, basados en ontologías comunes y estándares internacionales.

La falta de estándares para el intercambio de datos condujo a un entorno, en el cual cada firma debía realizar una integración a la medida de sus necesidades. Zuehlke [13] en el 2010 aconseja aplicar estándares en todos los niveles de la empresa manufacturera, tanto en lo referente a hardware como a software, a fin de reducir los esfuerzos de planificación y permitir reutilizar componentes. El autor también indica que la estandarización por sí sola no es suficiente y, advierte que, antes de emplear nuevas tecnologías extrapoladas de otros ámbitos debe certificarse fiabilidad y seguridad bajo condiciones industriales.

Shen *et al.* [2] indican que, luego de revisar cerca de 100 artículos publicados entre 1998 y 2006 (relacionados con unos 70 proyectos diferentes), se encuentra que muchos proyectos recientes se parecen demasiado a algunos completados antes de 1998 y que, la integración dentro de la empresa manufacturera continúa siendo un problema muy difícil de resolver. Esto es particularmente cierto en el caso de pequeñas y medianas empresas que cuentan con sistemas preexistentes, obsoletos, heterogéneos, los cuales hay que mejorar e integrar antes que descartar. Milagaia en el 2009 [4] ratifica este dato; la tecnología existente falló de proveer una solución completa, independiente de la plataforma y formato de datos. El autor postula que debe existir un resultado intermedio, en el cual coexistan sistemas preexistentes junto con aquellos que utilizan los nuevos arquetipos tecnológicos. A tal fin, paradigmas tales como los emergentes WS y SOA, ambos soportados por distintas plataformas independientemente de la tecnología utilizada, parecen tener muchas posibilidades.

Shen *et al.* [2] señalan como otro tema difícil, aún no resuelto, la integración completa de control y planificación de procesos manufactureros, particularmente porque está en juego la información en tiempo real obtenida por sistemas de colección de datos, problema nada trivial. Debe establecerse el mapeo correcto para conectar los bits de información de distintos sistemas, con diferentes identificadores, representaciones de datos y niveles de granularidad: desde segundos (o en algunos casos milisegundos) a meses y, de equipamientos aislados al manejo de puntos de producción completos.

En Vernardat [15] se citan algunas otras causas de falla de los sistemas CIM: están basados en exceso en estándares específicos o propietarios, se supone operación sincrónica de todos los sistemas (tienen que ejecutarse todos al mismo tiempo) y, su ampliación se complica cuando el número de componentes interconectados es significativo.

Los primeros sistemas CIM centrados en la tecnología fallaron y, se intentó mejorarlos tomando en cuenta factores humanos. En 1993 la CASA publica un modelo revisado que trataba de enfocarse más en los procesos de negocios, intentando reflejar las tendencias vislumbradas en la economía global y, por tanto, se coloca al

cliente en su centro. Actualmente las arquitecturas SOA se centran en los procesos de negocios antes que en la tecnología. Zuehlke [13] en el 2010 aconseja: no olvidar las lecciones aprendidas en la era CIM: ¡cualquier sistema técnico que se diseñe debe tener al ser humano como centro! Una fábrica carente de personas es una aberración, los sistemas futuros tendrían que enfocarse en las personas y sus capacidades, no los humanos ajustarse a cualquier tecnología que surja. También hay que recordar a Harrington, el primero en proponer el concepto CIM, que en su momento manifestó: CIM no significa una fábrica totalmente automatizada; por el contrario, todo el personal tiene que estar muy involucrado en todas sus áreas.

### 3.3 SOA a nivel de dispositivos de planta

En el 2004 Microsoft y algunas empresas de impresoras, extienden la idea de WS a dispositivos embebidos de bajo nivel al proponer un subconjunto de estándares conocidos como *Device Profile for Web Service* (DPWS). Además de los WS básicos del estándar se incluye *WS-Discovery* para descubrir servicios de forma “*plug-and-play*” y *WS-Eventing* para notificación de eventos asincrónicos en modo *publish-subscribe*, permitiendo que dispositivos que usan WS puedan comunicarse directamente entre sí (*peer-to-peer*). La identificación de un nuevo dispositivo puede hacerse sin la necesidad de realizar muchas modificaciones en el sistema en su conjunto, hecho muy importante principalmente en el caso de las Pymes, donde las inversiones en las TIC se evitan debido al tema de la escalabilidad. En [1] se realiza una descripción detallada de DPWS y las especificaciones de los protocolos incluidos en su *stack*.

Varios proyectos europeos [7] como SIRENA, SODA y SOCRADES se subieron a este tren, extendiendo las especificaciones a un *stack* de protocolos livianos que, puede portarse directamente a dispositivos embebidos con limitados recursos computacionales, habilitándolos a participar proactivamente en un entorno SOA y comunicarse mediante mensajes que adhieren al protocolo SOAP. Actualmente el *stack* DPWS, implementado por primera vez dentro del proyecto SIRENA por la compañía Schneider, está disponible como software *open source* (gSOAP) en dos versiones: C y Java.

DPWS es una tecnología de base para comunicación de dispositivos que puede fácilmente extenderse y componerse con otras especificaciones y tecnologías. En 2009 Oasis (*Organization for the Advancement of Structured Information Standards*) estandarizó oficialmente WS-DD, un conjunto de 3 estándares relacionados: DPWS, *WS-Discovery* y SOAP sobre UDP. Dicho estándar debería complementarse con definiciones de interfaces de servicios para un conjunto básico de los dispositivos más usados en piso de planta, favoreciendo la reutilización componentes en diferentes sistemas.

Una tecnología que merece especial atención en el campo de la automatización es OLE para control de procesos (u OPC, por *Ole for Process Control*), actualmente la solución más utilizada para intercambio de datos a nivel de dispositivos y aplicaciones de más alto nivel, tales como sistemas SCADA. La última versión de OPC, OPC-UA (UA, por *Unified Architecture*), agrega WS y, por tanto se prevé

facilitará la integración vertical de datos de automatización con aquéllos de procesos de negocios [16]. El uso combinado de DPWS y OPC-UA, debido a las similitudes en los componentes básicos de ambas especificaciones, proveerá un marco muy rico para SOA a nivel dispositivos; OPC-UA podría actuar como una posible extensión del *stack* DPWS existente [16]. Una de las ventajas más importantes de OPC-UA es la definición de modelos genéricos que proveen una forma estándar de describir y acceder a información de un dispositivo, eliminando la necesidad de programar WS a medida para cada uno de ellos. Esta característica sería útil para obtener una vista dual de los recursos utilizados, a través de OPC-UA y *WS-Management* (complemento u extensión de DPWS).

REST (por *Representational State Transfer*) es el principio arquitectural que se encuentra en el corazón de la Web y comparte los mismos objetivos que DPWS [17]. Las implementaciones REST se basan en el uso de HTTP como protocolo de aplicación directamente, en contraste con las aplicaciones SOAP WS que lo utilizan como protocolo de la capa de transporte. El aspecto ligero de REST lo hace el candidato ideal para dispositivos con recursos limitados que usualmente ofrecen funcionalidades más bien simples y atómicas (por ejemplo, leer el valor de un sensor inalámbrico). DPWS y REST no son contradictorios: DPWS puede implementarse en una aplicación con el estilo arquitectural REST.

El código que implementa la funcionalidad de un dispositivo y los servicios ofrecidos se genera mediante una *toolkit* DPWS, el *stack* TCP/IP usualmente lo provee la plataforma del dispositivo. El *stack* DPWS incluye la *toolkit* gSOAP con licencia GPL para construir WS en C/C++ y que, ofrece un entorno de desarrollo que usa el concepto de *plug-in* [18]. Contiene un compilador RPC que produce código intentando limitar el tamaño de la aplicación WS, reducir espacio de memoria como así también, requerimientos de comunicaciones y procesamiento; además es portable a muchas plataformas incluyendo dispositivos móviles.

Entre las ventajas más importantes de la incorporación de DPWS: la posibilidad de descubrimiento de fallas (en sensores, controladores, recursos, etc.) y su reparación en tiempo real; la información de bajo nivel y alta granularidad suministrada permite a los sistemas MES/ERP obtener indicadores claves de desempeño de la empresa, tales como el muy importante *Overall Equipment Effectiveness* (OEE) que mide la eficiencia productiva de la maquinaria industrial.

### **3.4 Integración de dispositivos preexistentes no habilitados para WS**

La integración de dispositivos de piso de planta basada en SOA, ya fue realizada a nivel de prototipo dentro del alcance del proyecto SOCRADES y, está en etapa de realización de pruebas para su evaluación en un ambiente manufacturero real [7]. Sin embargo, la implementación de WS en ellos tiene requerimientos en cuanto a capacidad de memoria, computacional y de almacenamiento que no son desdeñables. Este hecho constituye una barrera en el corto plazo, en el caso de dispositivos con tecnologías más viejas y con ciclos de vida en algunos casos de décadas, como los que aún suelen encontrarse en la mayoría de los pisos de plantas manufactureras. Aún si los dispositivos tienen recursos para ejecutar WS en forma nativa, el permitirlo

puede no ser realmente razonable o no tener ventajas competitivas, tal como ocurre con aquellos dispositivos dedicados a tareas muy específicas sin necesidad de capacidades avanzadas (sensores de proximidad, etiquetas de RFID, etc.).

Actualmente la integración de servicios “envolviendo” la funcionalidad de dispositivos preexistentes no habilitados para WS o, con recursos limitados, es difícil puesto que: utilizan protocolos de comunicación propietarios y/o mecanismos de intercambio de bajo nivel, típicamente para leer y escribir variables individuales de los mismos o, no están conectados en red y; cuando ello ocurre, no pueden interoperar y/o requieren entornos específicos para funcionar [19]. La transición hacia un piso de planta totalmente basado en SOA se realiza mediante *proxies* inteligentes, llamados *gateway* o mediadores de servicios en el proyecto SOCRADES. Los *proxies* (una PC industrial, dispositivos embebidos, etc.) conectados a un sistema de comunicación industrial implementan DPWS u OPC-UA y, además de los servicios ofrecidos por el o los equipamientos de planta “envueltos” mediante capas de software, se oculta el efecto de red necesario [19].

El reemplazo del equipamiento obsoleto debe ser gradual, debiendo además garantizarse la coexistencia de aquéllos no habilitados para WS con los nuevos que deberán proveer una interface dual: una habilitada para WS además de aquella relacionada con protocolos de comunicación propietarios o específicos del vendedor. Como regla general [19]: los *proxies* deberían estar lo más cercano posible (en términos de distancia a la red) a los dispositivos reales y, al menor nivel factible en la jerarquía de dispositivos de piso de planta. Cuánto más bajo es el nivel en el cual un dispositivo es “envuelto”, mayor flexibilidad se obtiene al participar en la composición de WS. El nivel más bajo posible en un escenario típico de una Pyme manufacturera sería el del PLC (suponiendo que no posea adaptador de red), el cual podría añadir a su parte de control en tiempo real y cíclica, una segunda que ofrezca el *stack* WS. Sin embargo, la integración a tan bajo nivel es muy costosa e implicaría rediseñar o introducir cambios significativos en el PLC. En la práctica, es preferible añadir WS en un *gateway* implementado en algún equipo conectado al PLC: ya sea un microcontrolador que realice alguna función de control a más alto nivel o, la PC industrial que contiene el SCADA del PLC.

#### **4 Trabajos futuros**

Una primera etapa de trabajo futuro se focalizará en probar, la adaptabilidad de herramientas disponibles para la construcción de software, que permita a un microcontrolador exponer el acceso a entradas y salidas de campo vía WS, tomando como referencia el estándar DPWS. En una etapa posterior se investigará la capacidad de complementar o extender DPWS, de forma tal de soportar las especificaciones OPC-UA y REST. En paralelo se lleva a cabo una encuesta y búsqueda de un caso de uso en un Pyme manufacturera de la zona de Rosario, Santa Fe.

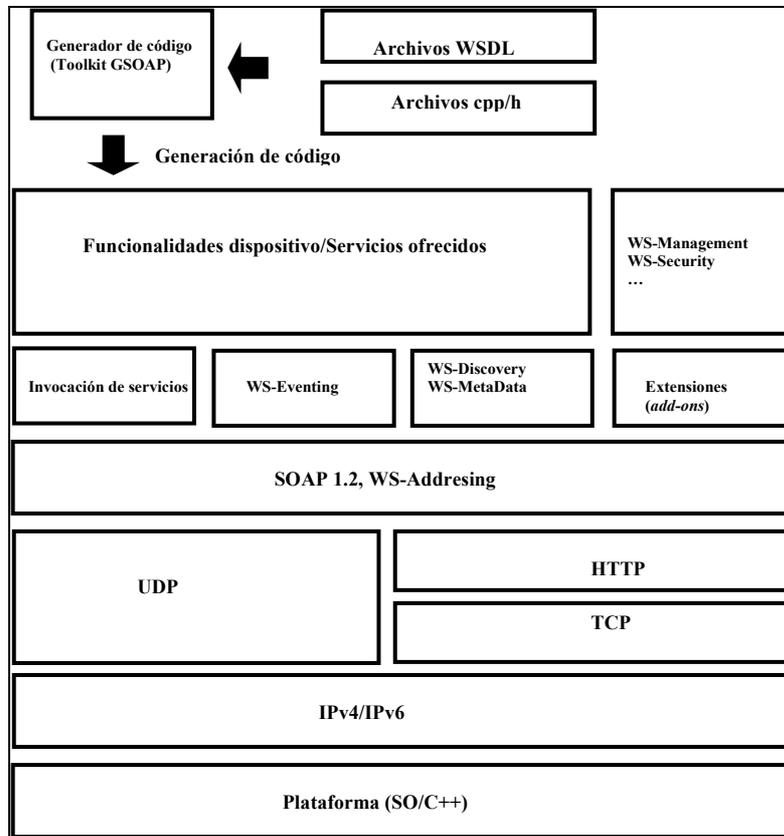


Fig. 1. Stack de software del dispositivo a utilizar en trabajos futuros

Dadas las restricciones de recursos habitualmente presentes en los sistemas embebidos centrados en microcontroladores, particularmente la memoria RAM disponible y el almacenamiento no volátil, se consideraron para el diseño alternativas que prioricen la optimización de recursos locales al dispositivo, en este caso la elección recayó en la implementación SOA4D DPWSCore (C DPWS toolkit), publicada por Schneider Electric SA, liberada bajo licencia GPL [20] (Fig.1).

El kit elegido es una tarjeta de desarrollo M52233DEMO con un microcontrolador de 32 bits ColdFire de Freescale [21], sus características salientes son:

- Core V2 ColdFire de microarquitectura a 60Mhz de Bus clock.
- Ethernet Media Access Controller (EMAC) 10/100 Mbps integrado en el chip.
- Puertos USB BDM Integrado y RS-232 serial, 4 LEDs programables.
- Acelerómetro triaxial, 2 push switches, potenciómetro de 5K  $\Omega$  con enable.
- 256 K bytes de memoria FLASH de programa/ 32K bytes de memoria RAM.
- TCP / IP (TCP/IP\_Lite) y MQX RTOS incluido en el TCP/IP\_Lite.
- Entorno de desarrollo y depuración Codewarrior 6.3.

El objetivo final es el desarrollo de una arquitectura de integración para Pymes manufactureras basada en SOA, que utilice ESB como columna vertebral [10] y sea compatible con ISA-95 [22], la cual será evaluada en un escenario demostrativo en la Pyme seleccionada como caso de estudio. Se encarará esta misión con un equipo interdisciplinario de ingenieros industriales, informáticos y electrónicos, perteneciendo los autores a este último grupo. Se seguirán los pasos típicos de un proyecto de integración conforme a ISA-95 [23]: en primera instancia se deberá documentar y comprender claramente las razones de la empresa para realizar la integración y, en una siguiente etapa, definir explícitamente los procesos de negocios y manufactureros involucrados, los cuales serán utilizados para el modelado usando lenguaje UML (por Unified Modeling Language); finalmente se realizará un mapeo de estos modelos orientados a objetos a otros basados en servicios [22].

## 5 Consideraciones finales

SOA en integración de aplicaciones industriales promete grandes beneficios, sin embargo, no existe aún sólida evidencia empírica de que todas las suposiciones positivas hechas acerca de SOA en este contexto, se justifiquen en todos los casos. Por tanto, en una Pyme manufacturera, la transición a este nuevo paradigma debe realizarse progresivamente, a fin de garantizar que las instalaciones de piso de planta existentes puedan seguir utilizándose durante este período, minimizando así la interrupción de las operaciones que realizan. Si se pretende obtener soluciones reutilizables y flexibles, no hay que olvidar que: SOA es una arquitectura de software y no sólo la implementación de una herramienta, lo cual implica entre otras cuestiones, disciplina en el cumplimiento de estándares y buenas prácticas y, quizás lo más importante y difícil, un cambio organizacional y cultural considerable.

## Referencias

1. Jammes, F., Smit, H.: Service-oriented Architectures for Devices - the SIRENA View. Proceedings of the 3rd IEEE International Conference on Industrial Informatics, pp. 140-147 (2005)
2. Shen, W., Hao, Q., Yoon, H.J., Norrie, D.H.: Applications of agent-based systems in intelligent manufacturing: An updated review. Advanced Engineering Informatics, Vol. 20, No.4, pp. 415-431 (2006)
3. Yoguel, G., Novick, M., Milesi, D., Roitter, S., Borello, J.: Información y conocimiento: la difusión de las tecnologías de información y comunicación en la industria manufacturera argentina. CEPAL, No. 82, pp. 139-156 (2004)
4. Milagaia, R.: DPWS Middleware to support Agent-based Manufacturing Control and Simulation. M.A. thesis, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (2008)
5. Sauter, T.: The Continuing Evolution of Integration in Manufacturing Automation. IEEE Industrial Electronics Magazine, Vol. 1, No. 1, pp. 10-19 (2007)
6. Emerson, D., Kawamura, H., Matthews, W.: Plant-to-Business (P2B) Interoperability Using the ISA-95 Standard. Yokogawa Technical Report, Vol. 43, pp. 17-20 (2007)

7. Nguyen, D.K., Savio, D.: Exploiting SOA for Adaptive and Distributed Manufacturing with Cross Enterprise Shop Floor Commerce. The 10th International Conference on Information Integration and Web-based Applications Services (WAS'2008), pp. 318-323 (2008)
8. Mathes, M., Gärtner, J., Dohmann, H., Freisleben, B.: SOAP4IPC: A Real-Time SOAP Engine for Time-Constrained Web Services in Industrial Automation. Proceedings of the 17th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (Euromicro PDP), pp. 220-226 (2009)
9. Guinard, D., Trifa, V., Mattern, F., Wilde, E.: From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices. Harrison, M., Michahelles, F., Uckelmann, D. (Eds.) *Architecting the Internet of Things*, Springer, pp. 1-33 (2011)
10. Groba, C., Braun, I., Springer, T., Wollschlaeger, M.: A Service-Oriented Approach for Increasing Flexibility in Manufacturing. 7th IEEE International Workshop on Factory Communication Systems (WFCS 2008), pp. 415-422 (2008)
11. Thramboulidis, K., Doukas, G., Koumoutsos, G.: A SOA-based Embedded Systems Development Environment for Industrial Automation. *EURASIP Journal on Embedded Systems*, Vol. 2008, pp. 1-15 (2008)
12. Scheer, A.-W., Jost, W., Güngöz, Ö.: A Reference model for industrial enterprises. Fettke, P., Loos, P. (Eds.) *Reference Modeling for Business Systems Analysis*. Hershey, USA: Idea Group Publishing, pp. 166-181 (2007)
13. Zuehlke, D.: SmartFactory: Towards a factory-of-things. *IFAC Annual Reviews in Control*, Vol. 34, No. 1, pp. 129-138 (2010)
14. Blecker, T., Graf, G., Kaluza, B.: *Internet-Technologies in Manufacturing - Applications and Management*. Ph.D. thesis, Universität Klagenfurt, Institute of Business Administration, Austria (2005)
15. Vernadat, F.B.: Interoperable enterprise systems: Principles, concepts, and methods. *Annual Reviews in Control*, Vol. 31, No. 1, pp. 137-145 (2007)
16. Jammes, F., Smit, H., Mensch, A., Harrison, R., Kirkham, T.: Use of web services for next-generation automation systems. Council for the Central Laboratory of the Research Councils, <http://epubs.cclrc.ac.uk/work-details?w=49594> (2009)
17. Moritz, G., Zeeb, E., Prueter, S., Golasowski, F., Timmermann, D., Stoll, R.: Devices Profile for Web Services and the REST. 8th IEEE International Conference on Industrial Informatics (INDIN), pp. 584-591 (2010)
18. van Engelen, R.A., Gallivan, K.A.: The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks. Proceedings of IEEE CCGrid Conference, pp. 128-135 (2002)
19. Karnouskos, S., Bangemann, T., Diedrich, C.: Integration of Legacy Devices in the Future SOA-based Factory. 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM), pp. 2109-2114, (2009)
20. DPWS Core project, <https://forge.soa4d.org/projects/dpwscore/>
21. Coldfire M52233 Student Learning Kit de Freescale semiconductor, <http://www.freescale.com/>
22. Pfadenhauer, K., Kittl, B., Dustdar, S.: Shop Floor Information Management and SOA. Dustdar, S., Eder, J. (Eds.) *Business Process Management Workshops 2006*. LNCS, Vol. 4103, pp. 237 – 248. Springer, Berlin Heidelberg (2006)
23. Gifford, C.: *When Worlds Collide in Manufacturing Operations: ISA-95 Best Practices Book 2.0*, International Society of Automation (ISA), Research Triangle Park, (2011)