



UCA

Universidad
de Cádiz

Universidad de

Cádiz

Escuela Superior de Ingeniería

IV Jornadas Predoctorales de la ESI

11 y 12 de Diciembre de 2012



Libro de Actas



**IV Jornadas Predoctorales de la
Escuela Superior de Ingeniería**

Libro de Actas



Cádiz, 11 y 12 de Diciembre de 2012

Editores: Juan José Domínguez Jiménez
Arturo Morgado Estévez
Inmaculada Medina Bulo

Imprime: Copistería San Rafael
C/ Benjumeda, 36
11003-Cádiz

ISBN-13: 978-84-695-6550-6

COMITÉ CIENTÍFICO

F. Javier Botana Pedemonte
Juan José Domínguez Jiménez
Pedro L. Galindo Riaño
Antonio J. Gámez López
Elisa Guerrero Vázquez
Manuel Jesús López Sánchez
Mariano Marcos Bárcena
Inmaculada Medina Bulo
Arturo Morgado Estévez

Guadalupe Ortiz
Andrés Pastor Fernández
Joaquín Pizarro
Mercedes Ruiz Carreira
Daniel Sánchez Morillo
Juan Manuel Doderó Beardo
Manuel Palomo Duarte

EDICIÓN TÉCNICA

Fernando Pérez Peña
Carlos Rodríguez Cordón

Arturo Morgado Estévez
Manuel Matías Casado

ORGANIZA



**Escuela Superior de Ingeniería
Universidad de Cádiz**



IV

Jornadas Predoctorales

ÍNDICE

INGENIERÍA INFORMÁTICA

Simulación MSKMC de Crecimiento Epitaxial de Puntos Cuánticos <i>J. Abujas, J. Pizarro, P. L. Galindo</i>	7
Mejora del proceso software en la metodología en V aplicando trabajo en equipo y factores de personalidad: una propuesta desde la simulación <i>Germán Fuentes Landi, Mercedes Ruiz Carreira</i>	11
Integración y extensión de lenguajes de marcado en el diseño de procesos de aprendizaje <i>José Miguel Mota Macías, Juan Manuel Dodero Beardo y Manuel Palomo Duarte</i>	15
WikiPapers, una recopilación colaborativa de literatura sobre wikis usando MediaWiki y su extensión semántica <i>Emilio José Rodríguez Posada, Juan Manuel Dodero Beardo</i>	19
Evaluando la calidad de los mutantes de primer orden producidos por GAmEraHOM <i>Emma Blanco Muñoz, Juan José Domínguez Jiménez e Inmaculada Medina Buló</i>	23
Nuevo enfoque para demostrar la efectividad de algoritmos naturales <i>L. Gutiérrez Madroñal, J.J. Domínguez Jiménez e I. Medina Buló</i>	27
Aplicación de la Técnica de Prueba de Mutación Evolutiva a C++ <i>Pedro Delgado Pérez, Inmaculada Medina Buló y Juan José Domínguez Jiménez</i>	31
Hacia la Unificación de Sistemas Multiagente con Arquitecturas Orientadas a Servicios en Sistemas Distribuidos de Fabricación <i>Antonio García Domínguez, Inmaculada Medina Buló, Mariano Marcos Bárcena</i>	35
Mejoras sobre el sistema DASICS de detección de ataques de seguridad que integra CEP y SOA 2.0 <i>Jose Antonio Dorado Cerón, Inmaculada Medina Buló, Guadalupe Ortiz</i>	39
Integrando el Internet de las Cosas y el Procesamiento de Eventos Complejos en las Arquitecturas Orientadas a Servicios <i>Juan Boubeta Puig, Guadalupe Ortiz, Inmaculada Medina Buló</i>	43

INGENIERÍA DE FABRICACIÓN

Propuesta metodológica para la evaluación del desgaste de herramientas por adhesión secundaria en el torneado en seco de aleaciones de aluminio empleando Microscopía de Foco Variable <i>D. García Jurado, A.J. Gámez López, M. Marcos Bárcena</i>	48
Propuesta de Metodología para la Evaluación de Material Adherido en Herramientas de Corte en Procesos de Torneado en Seco de Aluminio UNS A92024T3 Mediante Captura de Modelos 3D por Sistema de Variación de Enfoque y Perfilometría <i>José Manuel Mainé Torrejón, Mariano Marcos Bárcena, Lorenzo Sevilla Hurtado</i>	52

Análisis de las Desviaciones Microgeométricas en el Torneado en Seco de la Aleación de Al-Zn UNS A97075 <i>Francisco Javier Trujillo Vilches, Lorenzo Sevilla Hurtado, Mariano Marcos Bárcena</i>	56
Propuesta metodológica para el estudio de la interferencia tribológica entre el metal duro (WC-Co) y muestras de aleación de titanio (Ti6Al4V) tratadas por láser <i>J. M. Vázquez, F.J. Botana, M. Marcos</i>	60
Propuesta Metodológica para el Estudio de la Calidad del Taladrado de Materiales Compuestos de Fibra de Carbono con Matriz Epoxi <i>P. Mayuet, A. Lamikiz, M. Marcos</i>	64
La Gestión del Conocimiento y las Competencias de Comportamiento <i>M. Otero Mateo, A. Pastor Fernández</i>	68

INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Aplicación de cuestionarios médicos para la detección precoz domiciliar de exacerbaciones de EPOC <i>M.A. Fernández Granero, D. Sánchez Morillo, A. León Jiménez, F. Crespo Foix</i>	73
Control de estados de equilibrio inestables en sistemas dinámicos: Aplicación al control de sistemas caóticos <i>Manuel Prian Rodríguez, Manuel Jesús López Sánchez</i>	77

Integrando el Internet de las Cosas y el Procesamiento de Eventos Complejos en las Arquitecturas Orientadas a Servicios

Juan Boubeta Puig, Guadalupe Ortiz, Inmaculada Medina Bulo

Departamento de Ingeniería Informática, C/ Chile nº1 – 11002 Cádiz, 956015692, juan.boubeta@uca.es

Resumen

El Internet de las cosas proporciona una gran cantidad de datos que pueden ser compartidos por miles de organizaciones a nivel mundial. Integrando las arquitecturas orientadas a servicios con las dirigidas por eventos obtendremos arquitecturas apropiadas para la comunicación eficiente de dichos datos, proporcionados en formato de eventos por los distintos servicios de estas organizaciones. En este artículo, se propone el uso del procesamiento de eventos complejos en dichas arquitecturas para analizar estos eventos y así detectar situaciones relevantes en tiempo real. Además, se describe un caso de estudio para la detección de los distintos niveles de calidad del aire.

1. Introducción

El Internet de las cosas o *Internet of Things* (IoT) es un nuevo paradigma que está creciendo rápidamente en las telecomunicaciones. Se trata de una red interconectada de “cosas” u objetos, como etiquetas de radiofrecuencia, sensores u actuadores, identificados unívocamente a través de un esquema de direcciones, permitiendo que cada objeto interactúe y coopere con los demás objetos para alcanzar así un objetivo común [1]. Esta ingente cantidad de objetos proporcionados por plataformas IoT pueden considerarse como eventos simples que podrán ser procesados y correlacionados en tiempo real para beneficiarnos de la información que proporcionan; aunque es impensable gestionar manualmente todos estos eventos generados diariamente desde sistemas heterogéneos.

Para solventar este problema, en este artículo proponemos una arquitectura SOA 2.0 [2], esto es, una arquitectura orientada a servicios dirigida por eventos (ED-SOA), obteniéndose así las ventajas de ambas arquitecturas: una arquitectura orientada a servicios o *Service-Oriented Architecture* (SOA) es una solución eficiente para la implantación de sistemas en los que la modularidad y las comunicaciones entre terceros son un factor clave, desarrollándose así aplicaciones distribuidas compuestas de componentes (servicios) reutilizables y compartibles. Por otro lado, una arquitectura dirigida por eventos o *Event-Driven Architecture* (EDA) se caracteriza porque las comunicaciones se llevan a cabo por medio de eventos de una forma totalmente desacoplada, ya que los eventos son generados por los productores de eventos y serán los consumidores los que se encarguen de

interceptarlos y procesarlos. Con vistas a analizar continuamente toda la información que fluye por esta arquitectura ED-SOA, para detectar cuanto antes y de forma automática las situaciones que son críticas para los procesos de negocio, se integra con el procesamiento de eventos complejos o *Complex Event Processing* (CEP) [2], una tecnología que permite procesar y analizar grandes cantidades de eventos, así como correlacionarlos para detectar y responder en tiempo real a dichas situaciones. Para ello, se utilizan unos patrones de eventos que inferirán nuevos eventos más complejos y con un mayor significado semántico, que ayudarán a tomar decisiones ante las situaciones acontecidas.

Así pues, combinando SOA 2.0 y CEP podremos detectar eventos relevantes en sistemas complejos y heterogéneos. Esto requerirá un bus de servicios empresarial o *Enterprise Service Bus* (ESB) que actuará como capa de integración, reduciendo el acoplamiento.

El resto del artículo se estructura de la siguiente forma. En la Sección 2 se definen IoT y CEP. En la Sección 3 se propone una SOA 2.0 que integra IoT y CEP. En la Sección 4 se describe un caso de estudio para la detección de niveles de calidad de aire en distintas zonas geográficas. En la Sección 5 se enumeran algunos trabajos relacionados y, finalmente, se presentan las conclusiones y el trabajo futuro.

2. Conceptos previos

Internet de las cosas

El Internet de las cosas se define como una red formada por objetos interconectados que pueden ser identificados unívocamente y representados

en un mundo virtual. Para poder identificar estos objetos, será necesario el uso de mecanismos como identificación por radiofrecuencia. Algunos de los dominios donde puede ser aplicado IoT son [1]: transporte y logística, salud, entornos relacionados con hogares u oficinas, personal y social.

Cosm [3] es una plataforma IoT potente y escalable que gestiona cada día millones de datos provenientes de miles de personas, organizaciones y compañías a nivel mundial. Permite almacenar, compartir y extraer datos en tiempo real ofrecidos por los distintos objetos y dispositivos distribuidos geográficamente.

Además, Cosm permite establecer conexiones directas entre dos dispositivos, objetos u entornos, así como conexiones varios a varios. Así pues, podría desarrollarse, por ejemplo, una infraestructura que comunicase estaciones meteorológicas, monitores que analicen la calidad del aire y dispositivos de sensores móviles en la que cada uno de estos elementos podría comunicarse en tiempo real con el resto.

Cosm dispone de una buena documentación y API RESTful que proporciona los datos como flujos de datos en formato XML, CSV o JSON. Esta API permitirá que los datos puedan ser enviados a Cosm así como recibidos en tiempo real por las aplicaciones en cuestión.

Procesamiento de eventos complejos

CEP [2] es una tecnología que proporciona un conjunto de técnicas que ayudan a hacer un uso eficiente de las arquitecturas EDA. Se basa en el filtrado de eventos irrelevantes y en el reconocimiento de patrones de los eventos que sí son relevantes. Además, hace posible la detección de nuevos eventos más complejos, conocidos como “situaciones”; es decir, CEP permite la captura y correlación de cualquier tipo de evento con el fin de detectar situaciones de una mayor complejidad semántica e inferir conocimiento valioso para los usuarios finales.

La característica principal de estos eventos complejos procesados con CEP es que pueden ser identificados e informados en tiempo real, reduciendo la latencia en la toma de decisiones; a diferencia del software tradicional de análisis de eventos, que no funciona en tiempo real.

3. Propuesta

En este artículo, aplicamos nuestra arquitectura que integra CEP y SOA 2.0, definida en un trabajo previo [4], al Internet de las cosas. En concreto, utilizamos la plataforma IoT Cosm como productor de eventos (véase Figura 1).

La información proporcionada por Cosm será enviada al ESB Mule [5], haciendo uso de un conector HTTP. Además, este ESB se integrará

tanto con una BD NoSQL, que permitirá disponer de un histórico de eventos, como con el motor CEP Esper [6]. Este motor analizará y correlacionará los eventos para detectar las situaciones críticas o relevantes, a partir de los patrones definidos en dicho motor.

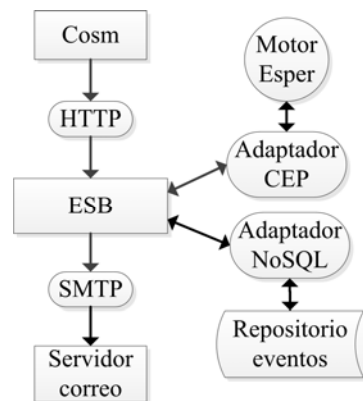


Figura 1. Arquitectura SOA 2.0 propuesta para la integración con IoT y CEP.

Una vez detectadas dichas situaciones se enviarán al ESB, que las transmitirá posteriormente a los consumidores de eventos interesados. En este caso, el consumidor de eventos será un servidor de correos, utilizándose para ello un adaptador SMTP.

4. Caso de estudio

Descripción

Actualmente existe una gran preocupación por la calidad de aire que respiramos cada día, debido a que una mala calidad del aire puede ser bastante perjudicial para nuestra salud.

Ante esta situación, la agencia de protección medioambiental de los Estados Unidos (EPA) ha definido un índice denominado *Air Quality Index* (AQI) [7] que informa diariamente sobre la calidad del aire de las zonas de los Estados Unidos, clasificada en niveles que oscilan desde valores “buenos” hasta “peligrosos” para la salud. Además, este índice especifica los problemas de salud que podrían aparecer en las próximas horas o días, en el caso de que una persona respire aire en malas condiciones.

El objetivo de este caso de estudio es demostrar que CEP es una solución efectiva y alternativa para detectar en tiempo real (minuto a minuto) los distintos niveles de calidad del aire en distintas zonas geográficas a nivel mundial; en comparación con la herramienta EnviroFlash [8] de la EPA, que proporciona dicha información hora a hora y solo para zonas pertenecientes a los Estados Unidos.

El uso de CEP permitirá a los responsables de salud estar al tanto en tiempo real de aquellas situaciones en las que la población se encuentre

respirando aire de mala calidad, para mitigarlas lo antes posible. Además, ayudará a que los ciudadanos tomen las precauciones necesarias para proteger la salud de sus familiares.

En este caso de estudio, los productores de eventos son sensores localizados en distintas zonas geográficas. En particular, hemos elegido los tres sensores *Weather Tunnel* disponibles en Cosm. Cada sensor dispone de un flujo de datos en Cosm donde se almacenará los datos (luz ambiente, sonido ambiental, temperatura, humedad, monóxido de carbono, dióxido de carbono y calidad del aire) tomados por el sensor cada 60 segundos.

Por otro lado, los consumidores de eventos podrán ser: *hospitales*, el personal sanitario necesitará conocer los niveles de calidad de aire identificados en cada zona, para tomar las medidas oportunas ante los pacientes afectados por la respiración de una mala calidad de aire; la *Organización Mundial de la Salud (OMS)*, que se suscribirá a los eventos que identifiquen los distintos niveles de calidad de aire a nivel mundial; y los ciudadanos, que se les informarán continuamente sobre la calidad del aire en sus respectivas zonas, recomendándoles además cómo proteger la salud de su familia.

Atendiendo a los requisitos reales para detectar los niveles de calidad de aire, hemos definido 7 patrones de eventos complejos (véase Tabla I).

Tabla I. Patrones de eventos de calidad del aire

Patrón Calidad Aire	Condiciones Aire
Bueno	≥ 0 y ≤ 50
Moderado	≥ 51 y ≤ 100
InsalubreParaGrupos	≥ 101 y ≤ 150
Insalubre	≥ 151 y ≤ 200
MuyInsalubre	≥ 201 y ≤ 300
Peligroso	≥ 301 y ≤ 500
CambioNivel	Cambio del nivel de calidad del aire en una determinada zona en el mismo día.

Implementación

A continuación, se describen los pasos seguidos para implementar este caso de estudio:

1º Integración de Cosm con Mule: se ha utilizado Mule Studio, un IDE para desarrollar, depurar y desplegar aplicaciones Mule que ofrece un editor gráfico *drag-and-drop* y un editor de código XML. Con esta herramienta hemos creado una aplicación Mule con los siguientes tres flujos correspondientes a los sensores que hemos seleccionado de Cosm: Kilpisjärvi Biological Station (<https://cosm.com/feeds/21544>), situado en Finlandia; Arizona State Univ (<https://cosm.com/feeds/26118>), situado en Arizona; Univ. of New S. Wales (<https://cosm.com/feeds/22587>), situado en Australia.

Estos flujos Mule se encargarán de recibir datos de Cosm cada minuto a través de un conector HTTP, transformar los datos recibidos en formato CSV a eventos *Plain Old Java Object (POJO)* y, finalmente, enviarlos al motor Esper para su procesamiento, así como almacenarlos en una base de datos MongoDB.

2º Detección y notificación de los eventos complejos: los patrones de eventos descritos anteriormente han sido implementados utilizando el lenguaje EPL de Esper por varios motivos. En primer lugar, la curva de aprendizaje no es elevada, ya que su sintaxis se aproxima bastante a la del lenguaje SQL ampliamente conocido a nivel mundial. Por otro lado, EPL soporta de forma nativa varios tipos de formato de eventos: objetos Java/.NET, *maps* y documentos XML. Además, permite la personalización del lenguaje así como del motor Esper, escrito en Java y de código abierto. Seguidamente, se muestra únicamente la implementación del patrón *CalidadAireBueno*:

```
@Name('CalidadAireBueno')
insert into NivelesCalidadAire
(tiempo, localización,
temperatura, calidadAire, nivel)
select buenNivel.tiempo,
buenNivel.localización,
buenNivel.temperatura,
buenNivel.calidadAire, 'bueno'
from pattern [every buenNivel =
EventoT(calidadAire >= 0 and
calidadAire <= 50)];
```

Las condiciones que deben cumplirse para detectar el nivel de calidad de aire *bueno* se especifican en la cláusula `from pattern`. El operador `every` recorrerá cada uno de los eventos de tipo *EventoT* que lleguen al motor y se les asignarán el alias *buenNivel* cuando se encuentren eventos que cumplan las siguientes condiciones: el valor del atributo *calidadAire* se encuentre comprendido entre 0 y 50. A continuación, se creará un evento complejo con los datos especificados mediante la cláusula `select` y se insertará en un nuevo flujo de eventos denominado *NivelesCalidadAire*.

Los eventos complejos serán capturados por unos componentes específicos, denominados *listeners*, que se encargarán de enviarlos al ESB.

3º Integración de Mule con el servidor de correos: una vez lleguen al ESB los eventos complejos informando de los niveles de calidad de aire en cada zona geográfica así como los cambios de niveles producidos a lo largo de un

día, se comunicarán por correo electrónico a todos los que estén interesados en recibirlos, tales como hospitales, OMS y ciudadanos. Para ello, se ha utilizado el protocolo SMTP y la API *javax.mail*. Además, en estos mensajes se podrán añadir recomendaciones a los ciudadanos según la calidad del aire que respiren. Por ejemplo, “reducir el tiempo en actividades al aire libre” en el caso de que la calidad de aire sea insalubre.

5. Trabajos relacionados

Existen trabajos recientes que proponen la integración de IoT y CEP en SOA 2.0.

Da et al. [9] proponen una arquitectura SOA que soporta servicios personalizados centrados en el usuario altamente escalables para IoT, haciendo uso de un ESB y el motor de reglas Drools.

Por otro lado, Walczak et al. [10] proponen un enfoque para comunicación máquina a máquina y procesamiento de datos en aplicaciones del Internet del futuro. En este caso, también utilizan el motor de reglas Drools.

En nuestra arquitectura propuesta utilizamos el motor CEP Esper, en lugar de un motor de reglas, ya que como afirman Chandy y Schulte [11] son más rápidos y eficientes. Además, lo conectamos con Mule, uno de los ESB de código abierto más populares en la actualidad, facilitando la integración con otras aplicaciones y arquitecturas ya existentes.

6. Conclusiones y trabajo futuro

Se ha propuesto e implementado una arquitectura SOA 2.0 capaz de gestionar, analizar y correlacionar una ingente cantidad de datos provenientes de una plataforma IoT con el objetivo de detectar en tiempo real situaciones críticas o relevantes para un determinado dominio. Esto ha sido posible gracias al uso de un motor CEP en dicha arquitectura y su integración con un ESB, que facilitará la conexión de los distintos productores y consumidores de eventos.

Esta arquitectura se ha aplicado a un caso de estudio para detectar en tiempo real los distintos niveles de calidad de aire que se respiran en distintas zonas distribuidas geográficamente a nivel mundial. En concreto, se han utilizado tres sensores localizados en Finlandia, Arizona y Australia cuyos datos se encuentran disponibles en la plataforma IoT Cosm. Nuestro sistema puede detectar niveles de calidad de aire minuto a minuto, mientras que las herramientas existentes en la actualidad, como por ejemplo EnviroFlash, informan hora a hora. Por tanto, podemos concluir que CEP es adecuado para este propósito.

Aunque se ha tomado como caso de estudio la calidad del aire, nuestro sistema podrá ser aplicado a la prevención de enfermedades u otros campos no relacionados con la salud. Por consiguiente, podemos asegurar que la integración de Cosm y Esper, haciendo uso de Mule, permite procesar eventos reales compartidos por miles de individuos, organizaciones y empresas de todo el mundo.

Como trabajo futuro completaremos nuestra arquitectura con la integración de otros servicios Web, aplicaciones y sensores/actuadores como productores/consumidores de eventos.

7. Agradecimientos

Este trabajo se enmarca dentro del ámbito científico del Proyecto MoDSOA (TIN2011-27242) bajo el Programa Nacional I+D+i del Ministerio de Ciencia e Innovación.

8. Referencias

- [1] L. Atzori, A. Iera, y G. Morabito, “The Internet of Things: A survey”, *Computer Networks*, vol. 54, nº 15, pp. 2787–2805, 2010.
- [2] D. C. Luckham, *Event Processing for Business: Organizing the Real-Time Enterprise*, 1.ª ed. Wiley, 2011.
- [3] “Cosm”. <https://cosm.com> [última consulta: 2012/11/01].
- [4] J. Boubeta-Puig, G. Ortiz, y I. Medina-Bulo, “An Approach of Early Disease Detection using CEP and SOA”, *Proc. 3rd International Conferences on Advanced Service Computing*, Rome, Italy, 2011, pp.143–148.
- [5] “Mule ESB”. <http://www.mulesoft.org> [última consulta: 2012/10/31].
- [6] “Esper”. <http://esper.codehaus.org> [última consulta: 2012/10/31].
- [7] EPA, “Air Quality Index – A Guide to Air Quality and Your Health”, U.S. Environmental Protection Agency, Research Triangle Park, NC, EPA-456/F-09-002, 2009.
- [8] EPA, “EnviroFlash”, <http://www.enviroflash.info> [última consulta: 2012/10/30].
- [9] Z. Da, C. Bo, Z. Yang, y C. Junliang, “Future Service Provision: Towards a Flexible Hybrid Service Supporting Platform”, *Proc. APSCC, Asia-Pacific*, 2010, pp. 226 – 233.
- [10] D. Walczak, M. Wrzos, A. Radziuk, B. Lewandowski, y C. Mazurek, “Machine-to-Machine communication and data processing approach in Future Internet applications”, *Proc. 8th CSNDSP*, 2012, pp. 1 – 5.
- [11] K. M. Chandy y W. R. Schulte, *Event Processing: Designing IT Systems for Agile Companies*. USA: McGraw-Hill, 2010.