

Aplicación de coreografía de servicios para el acceso y gestión de sensores de monitorización de la salud

JL. Bayo-Monton¹, A. Lizondo García¹, A. Martinez-Millana¹, C. Fernandez-Llatas¹, V. Traver Salcedo¹

¹ ITACA. Universitat Politècnica de València. Camino de Vera s/n. 46022, Valencia, España {jobamon, arligar, anmarmil, cfllatas, vtraver}@itaca.upv.es

Resumen

Los avances en la tecnología electrónica y en tecnologías de la información y comunicaciones constituyen la palanca de cambio en los nuevos métodos de provisión de servicios de salud. En el campo de la telemedicina, la miniaturización de dispositivos de cómputo y la comercialización de kits para monitorizar variables fisiológicas está promoviendo la creación de soluciones técnicas de muy fácil uso y fiabilidad. No obstante, este tipo de soluciones se diseñan de manera ad-hoc, limitando la escalabilidad, reutilización y flexibilidad de las funcionalidades de adquisición, procesado, comunicación y análisis. En este artículo presentamos un novedoso modelo de comunicación basado en el paradigma de arquitectura orientada a servicios para la reutilización, integración y escalado de sistemas de monitorización de la salud. Para ello, se describen los resultados obtenidos al aplicar la coreografía de servicios sobre un montaje basado en una placa de Arduino y un kit de sensores fisiológicos eHealth.

1. Introducción

Internet of the Things (IoT) hace referencia a la capacidad de gestionar de manera ubicua dispositivos, sensores y actuadores en un entorno multiconectado [1]. Éste concepto se ha extendido a múltiples ámbitos profesionales [2] y personales [3] representando en los últimos años un acusado crecimiento. El sector de la salud no ha quedado al margen del auge del IoT, en el cual ya existen múltiples aplicaciones y servicios que tienen el objetivo de mejorar la calidad de la asistencia sanitaria [4].

Dentro del sector salud, la telemedicina es un campo abonado para la expansión y mejora de las tecnologías IoT [5]. La monitorización remota, destacando los sistemas más accesibles y de bajo coste que permiten llegar a un mayor número de pacientes, y el uso de actuadores ambientales constituyen la vanguardia en la aplicación de este tipo de tecnologías [6].

Existen multitud de sistemas y arquitecturas para monitorizar personas a distancia, la literatura recoge aplicaciones que permiten almacenar los datos recogidos por los sensores comerciales en dispositivos avanzados (móvil, tablet u ordenador) para después generar informes que serán estudiados por los profesionales asistenciales [7]; otro tipo de aplicaciones, envían directamente la información recogida por los aparatos mediante Bluetooth o Wi-Fi a otros dispositivos [8] y otros que la envían directamente a un servidor para su almacenaje en la nube [9]. Estos sistemas ad-hoc permiten la accesibilidad de los

datos de manera remota, sin embargo adolecen de la flexibilidad necesaria para de integrar nuevas aplicaciones o funcionalidades para mejorar sus prestaciones en un entorno tan dinámico como el del IoT.

La arquitectura que proponemos para solucionar este problema de integración está basada en el uso del paradigma de la Arquitectura Orientada a Servicios (SOA) y el modelo de Coreografía de Servicios[10]. En la Coreografía la funcionalidad se obtiene mediante la integración de servicios auto contenidos que cooperan entre sí. Cada servicio aporta su funcionalidad al proceso global para obtener la funcionalidad completa. El uso de este tipo de arquitectura va a permitir la creación de un sistema distribuido que dote de flexibilidad y capacidad de adaptación a los sistemas de monitorización a distancia, independientemente del tipo de dispositivo, y de esta manera evitar la rigidez que plantean los sistemas ad-hoc.

2. Materiales y métodos

A continuación se detallan los componentes utilizados para la prueba de concepto del uso de la coreografía de servicios en el sector de la eSalud.

2.1. Hardware: plataforma E-Health Kit.

Para la adquisición de los datos de monitorización de variables fisiológicas de una persona se empleó la plataforma *E-Health Kit* diseñada por CookingHacks®. Este sistema se conforma por un módulo compatible con las placas Arduino y Raspberry Pi, a la cual se conectan sensores fisiológicos (hasta diez sensores diferentes). La unión de estos dos módulos, permite el desarrollo libre de aplicaciones médicas y biométricas [11]. En este artículo implementamos la coreografía de servicios empleando la placa Arduino UNO.

En el momento del estudio únicamente se disponía de los siguientes siete sensores, al estar los otros tres agotados: temperatura, SpO₂, GSR, posición, EMG, ECG y respiración

2.2. Software: Coreógrafo.

El principal reto para desarrollar una arquitectura distribuida es que sea fácilmente escalable, flexible a los cambios, evoluciones y expansiones, que permita la máxima reutilización del software y hardware de las primeras versiones.

En el campo de los procesos de negocio, múltiples organizaciones están trabajando en base al paradigma SOA para abordar los retos mencionados [12-14]. Dicho modelo ampliamente aceptado en otros sectores de las tecnologías de la información, comienza a coger fuerza en el desarrollo de plataformas orientadas al ámbito de la salud [15].

En base a las definiciones de SOA [12-14], los sistemas que implementa dicha arquitectura están compuestos por servicios y cada servicio se define como unidad de funcionalidad auto-contenida. Además el sistema se encargará de ofrecer mecanismos de comunicación entre los servicios. De esta forma la composición de servicios con funcionalidades concretas, es la que permite ofrecer la funcionalidad completa del proceso, maximizando la escalabilidad y la flexibilidad buscada.

Existen dos aproximaciones para trabajar con SOA: la orquestación de servicios y la coreografía de servicios. La orquestación de servicios es una aproximación centralizada en la que existe un servicio que se encarga de coordinar al resto para ofrecer la funcionalidad del proceso. En el caso de la coreografía, no existe ningún elemento central que coordine el proceso, cada servicio es consciente de su papel en el proceso y colabora con el resto para ofrecer la solución de manera conjunta, siendo esta última aproximación la más adecuada para sistemas distribuidos no centralizados [16].

Para el desarrollo que se presenta en este artículo se ha optado por utilizar la implementación del Motor de Coreografía realizada por el grupo SABIEN y que ha sido presentada con anterioridad en artículos previos [10,17]. Dicho motor implementa las características del modelo SOA y de Coreografía. Además dispone de facilidades extra para el desarrollo rápido de servicios, así como de servicios ya predefinidos: Servicios que permiten la comunicación enviando mensajes de coreografía mediante REST, TCP, WebService o MSQueue. O Servicios que conectan a sistemas externos como R, Matlab o Drools.

3. Resultados

En este estudio se propone el uso de sistemas de coreografía como elemento integrador que permita la creación de sistemas distribuidos y capaces de cumplir con el paradigma de IoT.

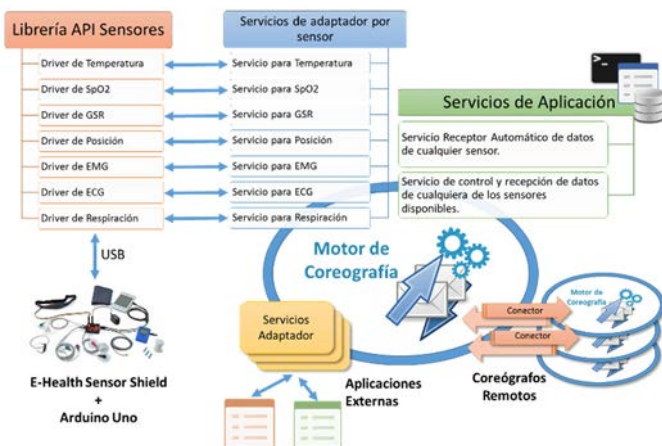


Figura 1. Arquitectura con el motor de coreografía como elemento central.

Para validar este concepto se propone una arquitectura (Figura 1) donde se utiliza como elemento central el motor de coreografía mencionado en materiales y métodos.

Dicho motor serviría como soporte para el desarrollo e integración de los servicios que permiten el control de los sensores, las aplicaciones y/o servicios que utilizan los datos de los sensores, así como de los servicios para utilizar los datos desde aplicaciones externas al coreógrafo. Además por la propia naturaleza del coreógrafo todos estos elementos pueden distribuirse en diferentes coreógrafos interconectados entre sí creando un sistema distribuido que puede ejecutarse en diferentes localizaciones.

Como prueba de la arquitectura descrita se ha realizado una implementación que consta de:

- Sensores: Sistema *E-Health Sensor Shield* conectado a una placa Arduino Uno y en la cual se cargará el software de control proporcionado por defecto por el fabricante.
- Librería *Application Programming Interface* (API): Librería que permite controlar cada uno de los sensores mediante comunicación serie por el USB.
- Servicios de los Sensores: Diferentes servicios para cada uno de los sensores definidos en la Librería API.
- Servicio Aplicación de Visualización: Servicio que es capaz de controlar los diferentes Servicios de los sensores y recibir los datos que estos proporcionan visualizándolos por pantalla.

En el caso del sistema de Sensores se ha optado por no modificar el software de control que proporciona el fabricante para validar la capacidad de integrar sistemas sin tener que adaptarlos.

La librería API y los Servicios de Sensores podrían haberse desarrollado como un único elemento, pero se han dividido en una librería que puede ser utilizada desde cualquier otro programa, de forma independiente del coreógrafo, y una librería de servicios de adaptación a cada sensor, para demostrar la capacidad del coreógrafo de integrar librerías de control ya preexistentes.

Cada uno de los servicios de los sensores comparte unos parámetros de configuración y comandos de control definidos previamente.

Los parámetros de configuración son los siguientes:

- COM: Nombre del puerto serie donde estará conectado el sensor por su comunicación.
- RunOnStart: Indica si el sensor debe iniciarse nada más crear el servicio, los datos capturados los enviará al destino definido en el parámetro "DataReceiver", empleando mensajes de coreografía de tipo evento.
- DataReceiver: Lista separada por punto y coma de los identificadores de los destinatarios a los que se enviarán los datos capturados.

Los métodos son los siguientes:

- `bool Measure(bool enable)`: Método que indica al sensor si debe empezar a capturar datos y enviarlos mediante eventos de coreografía o por el contrario debe para la captura. Los datos comenzarán a enviarse a los destinos indicados en el parámetro de configuración “DataReceiver”
- `bool MeasureToMe(bool enable)`: Método que indica los servicios que se desean suscribir a los destinos de envío por defecto o retirado de esta lista, dependiendo de si “enable” es verdadero o falso.

Mediante esta definición de los servicios de los sensores, es posible plantear diferentes servicios de aplicación que consuman los datos.

El primero que se ha realizado utiliza las bondades del motor de coreografía, el cual permite configurar los servicios desde un único fichero xml. Así pues se configuran los sensores con el parámetro *AutoStart* a verdadero y en el parámetro *DataReceiver* un valor que siga un patrón, por ejemplo “BiometricSensorData.TIPO_SENSOR.NUMERO” que aplicado al sensor de temperatura sería “BiometricSensorData.Temp.001”. De esta forma, un servicio que en su configuración indique que recibirá los mensajes con destino el patrón “BiometricSensorData”, recibirá los eventos de cualquier sensor del sistema o si queremos restringir solo a sensores de temperatura indicáramos “BiometricSensorData.Temp”.

La Figura 2 muestra un servicio que recibe automáticamente los eventos de cualquier sensor y los muestra en formato texto.

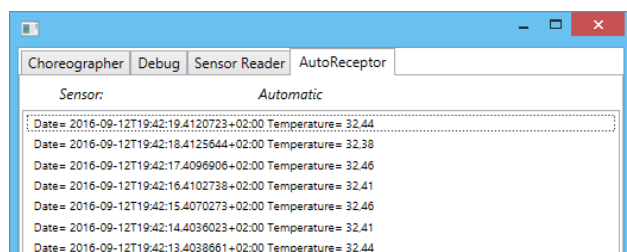


Figura 2. Servicio que recibe automáticamente los eventos de cualquier sensor y los muestra en formato texto.

Utilizando la estructura de los mensajes de coreografía, los datos son enviados como una lista de pares nombre del dato y valor del mismo. Como se aprecia en la Figura 2 los datos cuentan con un campo fijo de tipo marca de tiempo en que se tomó la medida y luego se concatenan el resto de valores.

Con este sencillo mecanismo de configuración del sistema, los sensores pueden iniciarse con la propia plataforma y cualquier servicio/aplicación puede en cualquier momento iniciarse y comenzar a consumir los eventos que desee.

El segundo de los servicios desarrollados es un servicio aplicación que permite desde la interfaz gráfica controlar los diversos servicios sensores mediante los métodos definidos anteriormente. Cuenta con un desplegable donde

seleccionar el sensor a controlar/leer y con una botonera que envía los diferentes comandos definidos.

Los botones “Start Sensor” y “Stop Sensor” envían un mensaje de coreografía al método “Measure” del sensor seleccionado, con el valor “enable” correcto.

Los botones “Subscribe to Sensor” y “Unsubscribe from Sensor” envían el mensaje de coreografía al método “MeasureToMe” del sensor escogido y con el valor “enable” apropiado.

Las respuestas de métodos solicitados al sensor son listadas en la parte inferior de la aplicación, así como la lectura de los eventos con los datos que captura el sensor en formato texto.

Los datos de la medida también son representados en formato de gráficas de evolución como demostración de una lectura semántica de los datos y no como texto sin analizar. En la Figura 4 podemos ver la captura de pantalla que ejemplifica el funcionamiento con los datos recibidos del sensor de temperatura.

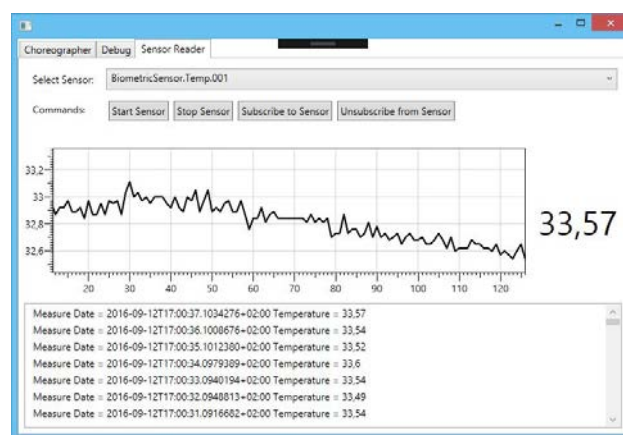


Figura 3. Funcionamiento de la aplicación con los datos recibidos del sensor de temperatura.

4. Discusión

En este artículo se ha presentado una arquitectura de comunicaciones basada en coreografía de servicios para el acceso a datos de salud de manera escalable y flexible. La contribución al estado del arte consiste en la capacidad del coreógrafo para abstraer la capa física (adquisición de los datos mediante sensores) y su encapsulamiento mediante servicios y anotaciones, permitiendo el acceso ubicuo, automático e independiente de la plataforma física a los datos capturados. Experiencias previas [5-6] empleaban arquitecturas ad-hoc basadas en el uso de placas de adquisición de datos orientadas a una determinada interfaz física (p.ej.: puerto serie RS-232) y al desarrollo de la lógica necesaria para acceder a los datos de interés. Por contra, la arquitectura presentada permite acceder a los datos de interés independientemente de la placa de adquisición y de la interfaz, ya que los canales lógicos de acceso a la información se encapsulan como servicios, pudiendo ser instanciados por agentes distribuidos y fuera del entorno del sistema de monitorización.

Para poner en valor la arquitectura propuesta, se ha implementado un prototipo con la placa Arduino y la

extensión de sensores de salud comercializada por CookingHacks® (eHealth kit), ampliamente utilizada por la literatura con sistemas desarrollados con la única finalidad de acceder a los datos de determinados sensores, y presentarlos a agentes conocidos (generalmente un servidor o una aplicación móvil). La solución propuesta permite que cualquier agente, independientemente de su implementación, tipo de tecnología y ubicación, pueda instanciar los servicios de captura de datos del montaje Arduino + eHealth kit.

El uso de un componente como el coreógrafo permite un grado de flexibilidad no observado en la literatura, ya que la arquitectura propuesta es capaz de trabajar con la extensión de CookingHacks®, y cualquier otra que proporcione datos de monitorización sin necesidad de modificar o recompilar. Entre las posibles aplicaciones de nuestra propuesta de arquitectura, destacamos la posibilidad de mejorar los esquemas de conexión distribuida a sensores de salud y sensores ambientales, y la posibilidad de implementar protocolos para el intercambio de datos clínicos que permitan la integración de los datos de salud adquiridos.

5. Conclusiones

La implementación de nuevos esquemas de comunicación basados en la coreografía de servicios en lugar de implementaciones ad-hoc permite la creación de nuevas aplicaciones de IoT flexibles y escalables en el sector de la salud.

Agradecimientos

El trabajo realizado ha sido posible gracias a la financiación de los proyectos EMBLEMA del instituto ITACA de la Universitat Politècnica de València.

Coautora Aroa Lizondo García: Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad: Programa Promoción de Empleo Joven e Implantación de la Garantía Juvenil 2014 (PEJ-2014-A-06813). La actividad subvencionada se enmarca en el Sistema Nacional de Garantía Juvenil y se cofinancian en el marco del Programa Operativo de Empleo Juvenil, con recursos financieros procedentes de la Iniciativa de Empleo Juvenil (IEJ) y del Fondo Social Europeo (FSE) para el periodo de programación 2014-2020.

Referencias

- [1] Wortmann, F., Flüchter, K. Internet of things. *Business & Information Systems Engineering*, vol 57, sup 3, 2015, pp221-224 (ISSN: 0937-6429)
- [2] Russo et al. Exploring regulations and scope of the Internet of Things in contemporary companies: a first literature analysis. *Journal of Innovation and Entrepreneurship* 4:11, 2015, (ISSN: 2192-5372)
- [3] Calvaresi D, Cesarini D, Sernani S, Marinoni M, Dragoni AF, Sturm A, Exploring the ambient assisted living domain: a systematic review. *J Ambient Intell Human Comput*, 2016, pp 1-19 (ISSN: 1868-5145)
- [4] Whitmore A, Agarwal A, Da Xu L. The Internet of Things-A survey of topics and trends. *Information Systems Frontiers*, vol 17, sup 2, 2016, pp 261-274.
- [5] Warren, S. Beyond telemedicine: Infrastructures for intelligent home care technology. *Actas del Pre-ICADI Workshop Technology for Aging, Disability, and Independence*. 2003. pp. 26-27.
- [6] Mečiak, V., Blaho, M., Mrafko, L., & Mudráková, T. Sensor-based platform E-Health connection with Matlab. *Actas del 23 Technical Computing*, Praga, 2015.
- [7] Zainee, N. M., & Chellappan, K. Emergency clinic multi-sensor continuous monitoring prototype using e-Health platform. *Actas del IEEE Conference on Biomedical Engineering and Sciences (IECBES)*, Kuala Lumpur, 2014, pp. 32-37.
- [8] Petrellis, N., Birbas, M., & Gioulekas, F. Precision and power issues in a medical sensor controller. *Actas del 19th Panhellenic Conference on Informatics (PCI'15)*, Athens, 2015, pp. 171-176.
- [9] Nachabe, L., Girod-Genet, M., ElHassan, B., & Jammam, J. M-health application for neonatal incubator signals monitoring through a CoAP-based multi-agent system. *Actas del 3rd International Conference on Advances in Biomedical Engineering (ICABME'15)*, Beirut, 2015, pp. 170-173.
- [10] A. Martinez-Millana, C. Fernandez-Llatas, L. Sacchi, D. Segagni, S. Guillen, R. Bellazzi, and V. Traver, From data to the decision: A software architecture to integrate predictive modelling in clinical settings. *Actas del 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, 2015, pp. 8161-8164.
- [11] Página web de CookingHacks®. e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi. <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> (Consultada: Septiembre 2016)
- [12] Página web de The Open Group, The SOA Work Group. <http://www.opengroup.org/getinvolved/workgroups/soa>. (Consultada: Septiembre 2016)
- [13] Página web de OASIS, Advancing Open Standards for the Information Society, OASIS SOA Reference Model Technical Committee. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm (Consultada: Septiembre 2016)
- [14] Página web de OMG, Object Management Group. <http://www.omg.org> (Consultada: Septiembre 2016)
- [15] S. R. Loya, K. Kawamoto, C. Chatwin, y V. Huser. Service Oriented Architecture for Clinical Decision Support: A Systematic Review and Future Directions. *Journal of Medical Systems*, vol. 38, n° 12, 2014, pp. 140 (ISSN: 1573-689X)
- [16] M. Autili, D. Di Ruscio, A. Di Salle, y A. Perucci. CHOReOSynt: Enforcing Choreography Realizability in the Future Internet. *Actas del congreso 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, New York, NY, USA, 2014, pp. 723-726
- [17] C. Fernández-Llatas, J. B. Mocholí, A. Moyano, and T. Meneu. Semantic Process Choreography for Distributed Sensor Management, *Actas del SSW*, 2010, pp. 32-37.