

Construyendo Perfiles Virtuales Mediante el Procesamiento de Eventos Complejos

Javier Berrocal¹, Juan Boubeta-Puig², Carlos Canal³, Jose Garcia-Alonso¹,
Juan M. Murillo¹, and Guadalupe Ortiz²

¹ Dpto. de Ingeniería de Sistemas Informáticos y Telemáticos, Universidad de Extremadura, España

{jberolm, jgaralo, juanmamu}@unex.es

² Dpto. de Ingeniería Informática, Universidad de Cádiz, España

{juan.boubeta, guadalupe.ortiz}@uca.es

³ Dpto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga, España
canal@lcc.uma.es

Resumen A medida que se incrementa el número de dispositivos inteligentes, el esfuerzo requerido para adaptarlos a las necesidades de cada usuario también crece. Asimismo, el proceso de adaptación de un dispositivo al contexto de un usuario es todavía un proceso muy manual. A pesar de que en los últimos años han surgido algunas propuestas centradas en obtener la información contextual de los usuarios para crear sus perfiles virtuales, se necesitan soluciones novedosas que permitan crear perfiles más completos, que sean utilizados por los dispositivos inteligentes para adaptarse automáticamente a las necesidades de sus usuarios, redundando en una mayor exactitud de la adaptación. En este artículo se propone la integración del modelo computacional *People as a Service* (PeaaS) con el procesamiento de eventos complejos (CEP) para la creación en tiempo real de perfiles virtuales complejos desde el propio dispositivo móvil y la compartición de estos como servicios para el resto de sistemas y dispositivos. Además, se evalúa esta integración en un caso de estudio sobre Alzheimer. Los resultados confirman que el uso de la tecnología CEP para la identificación de información contextual compleja es posible.

Keywords: Información Contextual, CEP, PeaaS, Asper, MDD

1. Introducción

El incremento en las capacidades de los dispositivos embebidos ha permitido el desarrollo de *Dispositivos Inteligentes*. Estos dispositivos normalmente están conectados a Internet y proporcionan una representación virtual de ellos en la red contra la que otros dispositivos pueden interactuar. Esta red de dispositivos inteligentes interconectados ha permitido el desarrollo de la Internet de las Cosas (*Internet of Things*, IoT) [13] y la Web de las Cosas (*Web of Things*, WoT) [15]. Uno de los principales objetivos de estos paradigmas es simplificar la vida de las personas haciendo que la tecnología trabaje para ellos.

Se estima que en el año 2020 habrá entre 50 y 100 billones de dispositivos inteligentes conectados a Internet [18]. Actualmente, con el objetivo de aumentar la usabilidad, el comportamiento de estos dispositivos y de los sistemas basados en WoT pueden adaptarse a las preferencias y al contexto del usuario, el cual puede además cambiar considerablemente a lo largo del tiempo.

Sin embargo, el proceso de identificación y adaptación de un sistema o dispositivo al contexto de un usuario es todavía un proceso muy manual. Debe ser el usuario el que establezca las nuevas preferencias y el nuevo contexto para cada dispositivo y sistema. Tener que configurar en el futuro todos los dispositivos que darán soporte a las actividades cotidianas requerirá un gran esfuerzo, que tendrá que ser duplicado cada vez que haya un cambio en el contexto. Esta configuración manual es aceptable cuando se interactúa con un conjunto pequeño de dispositivos; sin embargo, si este número se incrementa como está previsto, la carga de gestión de los dispositivos requerirá un esfuerzo que reducirá notablemente los beneficios proporcionados por estos sistemas.

Por lo tanto, se necesitan soluciones que permitan identificar el contexto y las preferencias de las personas, para que posteriormente sea aplicado sobre los dispositivos. Actualmente, existen varios trabajos que definen distintas propuestas para crear el perfil virtual de una persona [2], [17]. De hecho, los autores de este artículo incluso han propuesto un modelo computacional móvil, llamado *People as a Service* (PeaaS) [14], que convierte al teléfono móvil en el elemento central para capturar y mantener dicho perfil. Este perfil es posteriormente provisto como un servicio para que el resto de sistemas y dispositivos puedan obtener la información necesaria para adaptarse a las necesidades del usuario. Además, para asegurar la privacidad de los datos, uno de los pilares de PeaaS es que cada usuario pueda definir qué sistemas externos pueden acceder a la información contextual.

Estos perfiles normalmente contienen información de bajo nivel obtenida de distintas fuentes y sensores. Con el propósito de poder identificar información contextual de alto nivel y, así, crear perfiles virtuales más completos, en este artículo se propone la integración de PeaaS con el procesamiento de eventos complejos (*Complex Event Processing*, CEP). CEP es una tecnología que permite analizar ingentes volúmenes de datos para detectar situaciones de interés en tiempo real, gracias al uso de un software específico: un motor CEP. Concretamente, este motor CEP recibirá como entrada eventos sobre los cambios contextuales de una persona. Entonces, estos eventos serán analizados junto con la información histórica y, en base a determinados patrones de eventos predefinidos, será identificada la información de alto nivel.

El resto del artículo se estructura de la siguiente manera. La Sección 2 propone una arquitectura novedosa para integrar PeaaS con un motor CEP, y especifica cómo se hace uso de CEP para la obtención de información contextual compleja. La Sección 3 describe un caso de estudio sobre Alzheimer en el que se aplica dicha arquitectura al ámbito de la salud. La Sección 4 especifica los trabajos relacionados. Finalmente, la Sección 5 presenta las conclusiones y el trabajo futuro.

2. Integración PeaaS y CEP

En los últimos años estamos viendo un considerable incremento en las capacidades de almacenamiento y cómputo de los teléfonos móviles. Las últimas investigaciones demuestran que es posible utilizarlos para realizar tareas que anteriormente estaban relegadas a un PC o un servidor (como construir y almacenar el perfil virtual de un usuario [14] o la ejecución de motores CEP [12]). En esta sección se detalla la integración de un motor CEP en el modelo computacional PeaaS con el objetivo de poder crear perfiles virtuales más completos. Además, siguiendo la filosofía PeaaS, tanto el perfil como el motor se encontrarían en el dispositivo final, facilitando la escalabilidad del sistema. De esta forma, las aplicaciones instaladas en los dispositivos móviles podrán hacer uso de dichos perfiles para adaptar su comportamiento, y el comportamiento de otros dispositivos asociadas a ellas.

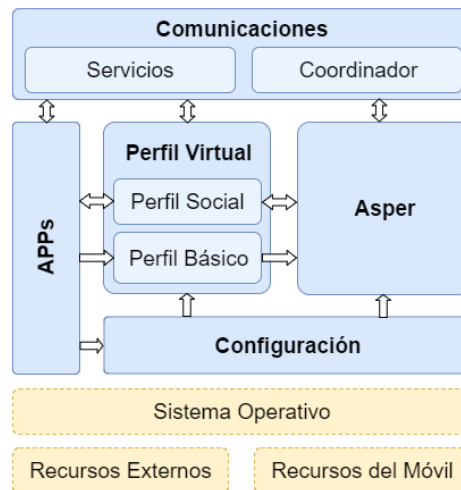


Figura 1. Arquitectura de PeaaS integrando el motor CEP.

La Figura 1 muestra una posible arquitectura de alto nivel de esta integración. Para mantener la filosofía de PeaaS, todos los módulos y responsabilidades de la arquitectura recaen sobre el dispositivo móvil. Así, el dispositivo móvil adquiere el rol principal en la arquitectura y se garantiza que el perfil virtual se encuentre únicamente almacenado en el dispositivo del usuario. Esto no significa que no se puedan utilizar otros sistemas externos, pero tienen un rol secundario en la arquitectura.

Debido a las restricciones de los sistemas operativos, la integración PeaaS-CEP está concebida para que sea implementada a nivel de aplicación. Por ello, todos sus módulos recaen sobre el sistema operativo para acceder a funcionalidades específicas y a los recursos del móvil o externos. Los *Recursos del Móvil* son

los distintos sensores del dispositivo (i.e., GPS, acelerómetro, sensor de luminosidad, etc.). Los *Recursos Externos* son sensores no incorporados en el dispositivo móvil pero con los que está conectado (pulseras cuantificadoras, sensores de pulso, beacons, etc.).

El elemento central de la arquitectura es el *Perfil Virtual*. Este módulo es el encargado de obtener y almacenar la información contextual del usuario. En concreto, este perfil puede obtener información de los recursos internos o externos conectados al móvil, de las aplicaciones que hacen uso de la arquitectura –por ejemplo, sobre cómo el usuario interacciona con ellas– o de servicios externos – como servicios meteorológicos, de tráfico, etc. Una particularidad de este módulo es que no sólo almacena la última y más actualizada información contextual, sino que mantiene un histórico con toda la información recopilada. Esta información es almacenada siguiendo un orden temporal, i.e. formando un *timeline*, de forma que también se pueden hacer inferencias y consultas basadas en la evolución y los cambios de un determinado dato en el tiempo. El perfil virtual es dividido en el *Perfil Básico* y el *Perfil Social*. El Perfil Básico contiene todos los datos en bruto obtenidos de los distintos recursos y servicios. El Perfil Social contiene el resultado del procesamiento de los datos en bruto y los eventos complejos obtenidos por el motor CEP.

Asper [11] es el motor CEP móvil propuesto para esta arquitectura. Este motor, una adaptación del motor CEP Esper para Android, recibe información del perfil virtual y del módulo de comunicaciones. De los perfiles básico y social obtiene los datos capturados de los distintos recursos y la información de alto nivel detectada con anterioridad. Del módulo de comunicaciones puede recibir información o eventos derivados de la invocación de distintos servicios desplegados en el móvil como, por ejemplo, la recepción de una notificación *push* con un determinado contenido en el mensaje. Toda la información obtenida es procesada por el motor para, primero, identificar información contextual de alto nivel –como cuál es la residencia habitual de una persona o su temperatura de confort– y, segundo, para la detección de si se cumplen determinadas condiciones que conlleven la ejecución de acciones –por ejemplo, si se detecta que un usuario llegará tarde a casa, se podría indicar al termostato que controla la temperatura de la vivienda del usuario para que retrase su encendido. Cuando el motor detecta información contextual de alto nivel ésta es almacenada en el perfil social, para que pueda ser accedida por cualquier aplicación. Cuando el motor detecta un patrón de eventos relativo a una acción que debe ser realizada, ésta es notificada a la aplicación, para que se dispere alguna funcionalidad o comportamiento, y/o al submódulo coordinador, en el caso de que sea necesario coordinar varias personas y dispositivos externos para completar la funcionalidad.

El módulo de *Comunicaciones* es el encargado de gestionar las interacciones con servicios y dispositivos externos no conectados directamente con el dispositivo móvil. Este módulo está dividido en dos submódulos: *Servicios* y *Coordinador*. El submódulo Servicios gestiona la comunicación con servicios externos. Este submódulo se relaciona con las aplicaciones basadas en esta arquitectura, proporcionándoles servicios de envío y recepción de notificaciones *push* basadas

en el contenido del perfil virtual; con el perfil virtual, para añadir información invocando servicios web; y con el motor CEP, para la redirección de notificaciones *push* requeridas para la identificación de patrones de eventos o para la invocación de determinados servicios como resultado de la ejecución de una acción tras la detección de un patrón. El módulo Coordinador es el encargado de coordinar distintos dispositivos cuando el motor CEP o una aplicación detecta que se debe realizar una acción en la que todos ellos están involucrados. Por ejemplo, si el motor CEP detecta que una persona tiene invitados en casa, se podría enviar un mensaje a todos ellos para saber cuál es su temperatura de confort y, así, posteriormente poder establecer en el termostato la temperatura adecuada.

El módulo de *Configuración* contiene información sobre los recursos y servicios de los que el perfil virtual tiene que obtener información, la frecuencia mínima con la que se debería obtener la información, y los patrones de eventos que tendrá que procesar el motor CEP. Esta información es proporcionada inicialmente al módulo de Configuración por cada una de las aplicaciones que hacen uso del perfil virtual, pero también pueden cambiar en tiempo de ejecución mediante la recepción de notificaciones *push* con dicha configuración. Así, por ejemplo, si en tiempo de ejecución se desea modificar los patrones de eventos, el experto del dominio podría utilizar cualquier ordenador para definir gráficamente nuevos patrones, o modificar los existentes, utilizando MEdit4CEP [6]. Una vez definidos, dichos patrones pueden ser incorporados a una notificación *push* y enviada al dispositivo de los usuarios de los que se desea adaptar el comportamiento. Esta notificación es recibida y gestionada por el módulo de Comunicaciones para actualizar los datos almacenados en el módulo de Configuración. El mismo proceso es utilizado para actualizar la frecuencia de obtención de información de los distintos recursos. El módulo de Configuración almacena y unifica la información de acceso a los sensores y los patrones de eventos.

Finalmente, el módulo *APPs* conforma las distintas aplicaciones que hacen uso del perfil virtual.

3. Caso de Estudio: SafeWalks y Asper

En esta sección se detalla un caso de estudio del uso de la integración PeaaS-CEP para la construcción de perfiles virtuales. SafeWalks [16] es una aplicación que monitoriza los movimientos de personas con una capacidad cognitiva reducida o en un estadio temprano de Alzheimer, pero que aún tienen cierta autonomía y viven en sus casas.

Esta aplicación monitoriza a estas personas creando un perfil virtual lo más completo posible. Esta información es posteriormente procesada para identificar las rutinas de una persona –i.e. a qué hora se levanta, a qué hora suele salir de paseo, cuál es el recorrido que suele realizar, cómo influyen las condiciones meteorológicas en dichas actividades, etc.– o si hay desviaciones de rutinas, es decir, que un usuario no esté realizando la tarea esperada, lo que podría indicar que está perdido. Cuando se identifican desviaciones se disparan distintas acciones para tratarlas; por ejemplo, notificando las desviaciones a los cuidadores

asociados con el usuario. Mediante la identificación de rutinas y desviaciones se consigue mejorar la seguridad y calidad de vida de los usuarios y de los cuidadores.

Esta aplicación se basa en el modelo computacional PeaaS. Para ello, hace uso de la implementación comercial nimBees [1]. Esta implementación cubre los módulos de Comunicaciones y Perfil Virtual de nuestra arquitectura descrita en la Sección 2. Sin embargo, en una primera versión de la aplicación, la detección de rutinas y de desviaciones fue directamente implementada mediante código Java. Así pues, en este caso de estudio se aborda la integración del motor CEP Asper en dicha aplicación y la migración de parte del código de detección de rutinas y desviaciones, con el objetivo de evaluar si es factible la utilización de estos motores para la identificación de información contextual de alto nivel.

La información contextual más importante captada por esta aplicación es: localización en el interior de la vivienda del usuario –para lo que se hace uso de Beacons [3]–, localización en exteriores –mediante el sensor de GPS del móvil–, las personas que acompañan al usuario –para lo que se utilizan las capacidades de comunicación de nimBees– y el clima –que se obtiene invocando a un servicio externo. Toda la información capturada es almacenada en un histórico ordenado según la hora del día, lo que permite identificar las rutinas diarias del usuario.

Las acciones más importantes del flujo de información para la construcción del perfil virtual del paciente son las siguientes: (1) el Perfil Virtual obtiene información de los sensores según la frecuencia definida por la configuración de la aplicación, (2) la información recuperada es almacenada en el Perfil Básico y enviada al motor CEP, (3) el motor CEP ejecuta los patrones definidos en la configuración para detectar eventos complejos, (4) si se detecta información de alto nivel, ésta es almacenada en el Perfil Social –por ejemplo, si se detecta una rutina que se está realizando, ésta se reforzaría– y (5) si se detecta una desviación de la rutina, ésta es notificada a los cuidadores mediante el uso de la plataforma nimBees.

Para que el motor CEP pueda detectar información de alto nivel o desviaciones de rutinas, se han definido una serie de patrones de eventos; por falta de espacio sólo se presentan dos a modo de ejemplo:

- **CambioRutinaParque:** este patrón comprueba cada día a las 10.00 h. si el usuario con Alzheimer que se encuentra solo en ese momento (sin cuidador) y, a pesar de que no está lloviendo, no está en el parque durante las próximas 2 horas. En ese caso, se enviará un correo electrónico alertando de esta situación tanto a su cuidador como a sus familiares. La Figura 2 muestra este patrón modelado gráficamente con la herramienta MEdit4CEP [6].
- **CambioRutinaBiblioteca:** este patrón comprueba cada día a las 10.00 h. si el usuario sin cuidador y, aunque está lloviendo, no está en la biblioteca durante las próximas 2 horas.

Cabe destacar que el uso de la herramienta MEdit4CEP para definir los patrones de eventos nos proporciona varias ventajas, entre las que destacan: (1) los expertos en Alzheimer pueden definir por sí mismos –sin ayuda de informáticos

ni expertos en CEP [5]– tanto las condiciones que deben cumplirse para detectar cambios de rutinas de los pacientes como las acciones que deben llevarse a cabo tras su detección, y (2) los modelos gráficos de estos patrones pueden transformarse automáticamente al código –correcto y libre de errores– desplegable en los motores CEP, gracias a las técnicas de desarrollo dirigido por modelos empleadas. Por lo tanto, si un experto en Alzheimer desea modificar algún patrón, para mejorar su precisión o adaptarlo a un comportamiento concreto, puede utilizar MEdit4CEP para definir gráficamente el nuevo patrón y transformarlo automáticamente a código, y la plataforma Web de nimBees para enviar notificaciones *push* a las aplicaciones de los pacientes sobre las que se desea aplicar los nuevos patrones.

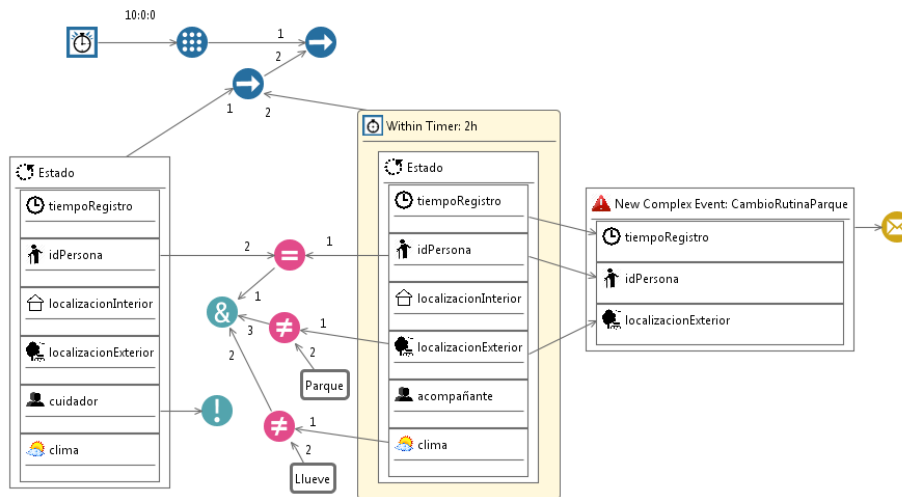


Figura 2. Patrón de eventos *CambioRutinaParque* modelado con MEdit4CEP.

Las pruebas iniciales realizadas sobre este caso de estudio nos han permitido comprobar que la utilización de motores CEP es una solución viable para la identificación tanto de información contextual compleja como de las acciones concretas que deben ejecutarse en cada momento. La utilización de estos motores permite incluir o modificar fácilmente los patrones de eventos para la detección en tiempo real de dicha información compleja. No obstante, el uso de un motor CEP en el dispositivo móvil podría conllevar un consumo de recursos más elevado que la implementación de las reglas de forma *ad-hoc* usando los lenguajes de programación específicos de cada dispositivo. En trabajos futuros se comparará este rendimiento.

4. Trabajos Relacionados

Desde el surgimiento de la tecnología CEP, los motores CEP se han ejecutado normalmente en ordenadores de escritorio o bien en servidores más potentes con el propósito de detectar situaciones críticas o relevantes en tiempo real para un dominio en particular. Con el avance de la tecnología, los dispositivos móviles son cada vez más potentes, lo que posibilita la aplicación de CEP en estos dispositivos.

Dunkel et al. [10] han denominado *mobile CEP* a este enfoque emergente en el que se hace uso de la tecnología CEP en *smartphones*. Según los autores, este enfoque, adoptado en nuestro trabajo, dispone de dos características principales: (1) conciencia de la situación, mediante la correlación de datos provenientes de los distintos tipos de sensores localizados en los propios teléfonos inteligentes y (2) conciencia del contexto, a través de la incorporación tanto de datos del contexto, como de dispositivos periféricos. Como ventajas significativas de este enfoque, destacan las siguientes: (1) procesamiento de los datos de los sensores sin necesidad de establecer ninguna comunicación con el servidor, privacidad de la información y correlación de datos en el propio dispositivo. En este trabajo, aplican este enfoque a un escenario de vida cotidiana asistida por el entorno (*Ambient Assisted Living, AAL*), concretamente implementan una app Android para la detección automática de caídas de personas ancianas. Mientras los autores definen los patrones de eventos mediante las *condiciones* que deben cumplirse para detectar una situación y el tipo de evento complejo (denominado *action*) que debe crearse tras su detección, en nuestro trabajo definimos los patrones de eventos como *condiciones*, *tipo de evento complejo* y *acciones* conforme a nuestros lenguajes específicos de dominio Model4CEP [7]. De esta forma, separamos la definición del propio patrón de las posibles acciones que deberían llevarse a cabo tras su detección como, por ejemplo, el envío de un correo electrónico o la invocación de un servicio.

Bruns et al. [8] también proponen el uso de CEP en el móvil para coordinar la gestión eficiente del personal de ambulancias ante un accidente de tráfico. Como novedad, utilizan además una máquina de estados finitos (*Finite State Machine, FSM*) para representar los estados de operación de una ambulancia. Gracias a la integración del motor CEP Asper [11] con FSM, se pueden detectar automáticamente cambios de dichos estados de operación. Este trabajo presenta la misma limitación del anterior en cuanto a la definición de los patrones de eventos.

Birth et al. [4] describen una EDA integrada con CEP y también implementan una app Android, haciendo uso de Asper, para proporcionar a los viajeros información en tiempo real sobre situaciones inesperadas que ocurran durante la ruta seleccionada por el usuario. La principal limitación de este trabajo es la dependencia de la arquitectura propuesta a este dominio de aplicación.

Finalmente, Coronado et al. [9] definen un enfoque basado en CEP distribuido para coordinar los servicios de automatización de tareas (*Task Automation Services, TAS*). Por un lado, los autores definen el concepto de “perfil” como el lugar (dispositivo móvil o servidor) en el que deben ejecutarse los patrones

de eventos. Estos perfiles se clasifican en 3 tipos: *web-driven execution* (ejecución de patrones en el servidor), *device-driven execution* (en el dispositivo) o *mixed execution* (distribución de perfiles entre el servidor y el dispositivo). Con respecto a la definición de patrones, como ya se ha mencionado, nosotros proporcionamos una herramienta de modelado gráfica (MEdit4CEP) que permite a los expertos en el dominio la definición de los mismos. Estos autores también implementan una app Android integrada con el motor Asper. Sin embargo, el servidor de coordinación envía (*push*) las notificaciones a los *smartphones* usando *Google Cloud Messaging* (GCM), mientras en nuestro trabajo se lleva a cabo con la plataforma nimBees, una solución que se ajusta mejor a las necesidades de los sistemas adaptados al contexto, ya que permite enviar notificaciones que pueden tener asociadas condiciones basadas en el contenido del perfil virtual del usuario.

5. Conclusiones y Trabajo Futuro

El aumento de las capacidades de los teléfonos móviles permite crear aplicaciones que puedan realizar funcionalidades cada vez más complejas. Actualmente, los usuarios están demandando que dichas funcionalidades sean capaces de cubrir sus necesidades con la menor configuración posible. La consecución de este objetivo será crucial para el éxito de muchas aplicaciones. En este artículo se integra el modelo computacional PeaaS con un motor CEP con la finalidad de crear perfiles virtuales más completos y de facilitar la identificación de situaciones contextuales complejas que conllevan a las aplicaciones a la ejecución de acciones.

Puesto que el consumo de recursos, y especialmente la batería, es especialmente sensible para las aplicaciones y los teléfonos móviles, actualmente estamos trabajando en evaluar el impacto que conlleva el uso de motores CEP para la identificación de información contextual compleja.

Agradecimientos. Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (proyectos TIN2014-53986-REDT, TIN2014-53555-REDT, TIN2015-67083-R, TIN2015-69957-R y TIN2015-65845-C3-3-R), la Junta de Extremadura (GR15098) y los fondos FEDER. Boubeta-Puig agradece la hospitalidad recibida durante su estancia de investigación con el Grupo Quercus de Ingeniería del Software de la Universidad de Extremadura, donde parte de este trabajo ha sido desarrollado.

Referencias

1. nimBees. <http://www.nimbees.com>, [Online]
2. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggle, P.: Towards a better understanding of context and context-awareness. In: Int. Symposium on Handheld and Ubiquitous Computing. pp. 304–307. Springer-Verlag (1999)

3. Apple Inc: ibeacon for developers. <https://developer.apple.com/ibeacon/>, [Online]
4. Birth, O., Hoffmann, M., Strassberger, M., Roor, R., Schlichter, J.: Concept for an Intermodal Traveller Information System with Real-Time Data Using Complex Event Processing. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC). pp. 2293–2298 (Sep 2015)
5. Boubeta-Puig, J.: Desarrollo Dirigido por Modelos de Interfaces Específicas de Dominio para el Procesamiento de Eventos Complejos en Arquitecturas Orientadas a Servicios. Ph.D. thesis, Universidad de Cádiz, Cádiz, España (Jul 2014), <http://rodin.uca.es:80/xmlui/handle/10498/17554>
6. Boubeta-Puig, J., Ortiz, G., Medina-Bulo, I.: MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0. *Knowledge-Based Systems* 89, 97–112 (Nov 2015)
7. Boubeta-Puig, J., Ortiz, G., Medina-Bulo, I.: Model4CEP: Graphical domain-specific modeling languages for CEP domains and event patterns. *Expert Systems with Applications* 42(21), 8095–8110 (Nov 2015)
8. Bruns, R., Dunkel, J., Billhardt, H., Lujak, M., Ossowski, S.: Using Complex Event Processing to support data fusion for ambulance coordination. In: 2014 17th International Conference on Information Fusion (FUSION). pp. 1–7 (Jul 2014)
9. Coronado, M., Bruns, R., Dunkel, J., Stipkovic, S.: Context-Awareness in Task Automation Services by Distributed Event Processing. In: Benatallah, B., Bestavros, A., Catania, B., Haller, A., Manolopoulos, Y., Vakali, A., Zhang, Y. (eds.) *Web Information Systems Engineering – WISE 2014 Workshops, Lecture Notes in Computer Science*, vol. 9051, pp. 190–203. Springer International Publishing, Switzerland (2015)
10. Dunkel, J., Bruns, R., Stipkovic, S.: Event-based smartphone sensor processing for ambient assisted living. In: 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS). pp. 1–6 (Mar 2013)
11. Eggum, M.: Asper - Esper for Android (2014), <https://github.com/mobile-event-processing/Asper>, [Online]
12. Eggum, M.: Smartphone Assisted, Complex Event Processing. Enablement of Esper on Android-based devices (2014), <https://www.duo.uio.no/bitstream/handle/10852/41663/Marcel-Eggum---Thesis.pdf?sequence=1&isAllowed=y>, Master Thesis [Online]
13. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29(7), 1645 – 1660 (2013)
14. Guillen, J., Miranda, J., Berrocal, J., Garcia-Alonso, J., Murillo, J.M., Canal, C.: People as a Service: A mobile-centric model for providing collective sociological profiles. *IEEE Software* 31(2), 48–53 (2014)
15. Guinard, D.: Towards the web of things: Web mashups for embedded devices. In: *In MEM 2009 in Proceedings of WWW 2009*. ACM (2009)
16. Lozano, P.P., Vereda, A.P., Murillo, J.M., Canal, C.: Safewalks: aplicación móvil de supervisión de pacientes de alzheimer. In: *Jornadas de Ciencia e Ingeniería de Servicios (JCIS)* (2015)
17. Makris, P., Skoutas, D.N., Skianis, C.: A survey on context-aware mobile and wireless networking: On networking and computing environments' integration. *IEEE Communications Surveys and Tutorials* 15(1), 362–386 (2013)
18. Perera, C., Liu, C.H., Jayawardena, S., Chen, M.: Context-aware computing in the internet of things: A survey on internet of things from industrial market perspective. *CoRR* (2015)