

Framework basado en smartphones para detección de ADL en entornos sensorizados simulados

Alejandro Pérez-Vereda *¹ [0000-0002-0195-0062], Luis Cabañero ¹ [0000-0002-2357-6710], Chris Nugent ² [0000-0003-0882-7902], Ian Cleland ² [0000-0003-2368-7354] and Ramón Hervás ¹ [0000-0001-9924-5443]

¹ Universidad de Castilla – La Mancha, Ciudad Real, España
{alejandropvereda, luis.cabanero, ramon.hlucas}@uclm.es

² Ulster University, School of Computing, Belfast, UK
{cd.nugent, i.cleland}@ulster.ac.uk

Abstract. Los entornos sensorizados son una de las principales contribuciones para mejorar la calidad de vida y promover la independencia de personas mayores. Dentro de este ámbito, la detección de la actividad del usuario es una característica fundamental para cualquier sistema de monitorización implantado en hogares inteligentes. En este artículo se detalla una propuesta basada en el uso de Avatares Digitales y Procesamiento Móvil de Eventos Complejos (más conocido en inglés como Mobile Complex Event Processing) para la monitorización del usuario y detección de situaciones de interés en tiempo real. Los smartphones son dispositivos pervasivos presentes en la vida del usuario y con suficientes capacidades de procesamiento y comunicación con el entorno; características que lo han convertido en un candidato idóneo para ser el principal conocedor del usuario e interfaz de este con su entorno asistido. Adicionalmente, acompañamos nuestra propuesta con una prueba de concepto fundamentada en el uso de Digital Twins con un simulador automático. Esta sirve de modelo capaz de facilitar el desarrollo y pruebas, además de reducir drásticamente los costes y dificultades asociadas al diseño de entornos asistidos sensorizados.

Keywords: Mobile Complex Event Processing, Activity Recognition, Ambient Assisted Living, Digital Avatars, Digital Twins.

1 Introducción

La esperanza de vida global ha aumentado más rápido que la esperanza de vida sana, siendo la primera de 73,3 años y la segunda de 63,7 años en 2019¹. Esto quiere decir que ha aumentado la proporción de años en los que las personas viven con algún problema de salud o discapacidad. Dicha situación se traduce en un problema grave para los servicios de salud que deben atender a zonas con una población envejecida, lo cual

¹ World health statistics 2022: monitoring health for the SDGs, sustainable development goals: <https://www.who.int/publications-detail-redirect/9789240051157>



requiere un gran gasto en recursos (desplazamientos, pequeños hospitales...) para poder satisfacer las necesidades sanitarias de este tipo de población.

A día de hoy, es posible encontrar soluciones para la monitorización a distancia de muchos aspectos de la vida de las personas como la nutrición, ejercicio, desplazamientos, etc. [1]. También existe mucha investigación en este ámbito, por ejemplo, para relacionar las actividades en el exterior y el uso del smartphone o las rutinas de sueño del usuario con la posibilidad de padecer depresión [2, 3]. Gracias a estas soluciones y a las herramientas de comunicación actuales, como videollamadas, es posible reducir los gastos en sanidad en lo relativo a la necesidad de monitorización de personas con requisitos sanitarios más acuciantes.

Adicionalmente, Ambient Intelligence [4] es una disciplina que trata de explotar el contexto de las personas que viven en un entorno “inteligente”, es decir, que presenta una sensorización y servicios que los hacen sensibles y capaces de actuar ante la presencia de usuarios. Concretamente, como resultado de aplicar Ambient Intelligence a ayudar a personas en edades avanzadas o con alguna enfermedad a llevar una vida sana e independiente por más tiempo, nacen las tecnologías relacionadas con Ambient Assisted Living (AAL).

AAL trata de mejorar la calidad de vida de estas personas haciendo de su hogar un entorno asistido e inteligente que monitorice sus necesidades y adapte sus servicios a ellas. Actualmente, en este tipo de soluciones el smartphone tiene un rol central, ya que se han integrado por completo en la vida diaria de los usuarios. Esto viene de la mano de mayores capacidades de detección, de almacenamiento y de cómputo de información contextual del usuario [5], convirtiéndose en el dispositivo más popular para el acceso a recursos y servicios.

Nuestra propuesta consiste en la aplicación del framework de computación móvil Digital Avatars y más específicamente de mobile Complex Event Processing (CEP) [6] para la monitorización de usuarios en hogares sensorizados. Digital Avatars [7, 8] es un framework basado en el modelo de People as a Service [9] que propone los smartphones como dispositivos pervasivos y con suficiente capacidad de computación y comunicación para hacer de interfaz del usuario con su entorno digital a través de la inferencia y uso de un perfil virtual (o avatar digital) que se ofrece como servicio de forma segura. Digital Avatars ofrece diferentes servicios a terceros para la interacción con información del usuario, siendo uno de ellos un motor CEP desplegado en el propio smartphone. Esta implementación de CEP permite a cualquier aplicación desarrollada para el framework obtener información del contexto del usuario utilizando diversas fuentes y sensores y analizar en tiempo real los datos para detectar situaciones de interés y reaccionar ante ellas. Por ello, consideramos esta propuesta como idónea para la monitorización y ayuda de usuarios de avanzada edad en sus hogares.

Concretamente, en este artículo basándonos en los trabajos previos mencionados [7, 8] donde se describe la arquitectura software Digital Avatars, se propone la aplicación del motor mobile CEP que permite la identificación de actividades relacionadas con las “Activities of Daily Living” (ADL). De forma complementaria, presentamos también un simulador basado en el concepto de Gemelos Digitales (Digital Twins en inglés) que permite diseñar de forma virtual una copia del entorno antes de implementarlo y simular el comportamiento habitual de un usuario dentro de él de forma auto-

mática. El simulador genera los mismos estímulos que recibiría el smartphone en el entorno real, pudiéndose usar nuestro mismo sistema CEP para probar el funcionamiento en dicho entorno. Este proceso de simulación y prueba usando CEP y el simulador automático permite minimizar los costos y mejorar el diseño de los entornos sensorizados en poco tiempo.

Para describir en más detalle el conjunto de Digital Avatars con el servicio CEP, el resto del artículo está organizado de la siguiente manera. La Sección 2 expone la arquitectura de Digital Avatars y muestra cómo dicho sistema es utilizado para la detección de situaciones de interés. La Sección 3 especifica el sistema de simulación desarrollado. Los trabajos relacionados a lo expuesto en este artículo son presentados en la Sección 4. Finalmente, la Sección 5 presenta las conclusiones y los trabajos futuros.

2 Digital Avatars Framework

Para exponer correctamente la propuesta, es necesario describir el framework Digital Avatars, así como sus componentes software. En esta sección, primero mostramos la arquitectura completa del framework para después focalizarnos en una de las principales contribuciones de este artículo que es la inclusión del motor CEP.

2.1 Arquitectura Software de Digital Avatars

La arquitectura de Digital Avatars se encuentra desplegada complemente en el smartphone y está desarrollada como una aplicación o servicio Android en segundo plano. En ella podemos diferenciar 3 capas diferentes: los recursos, una capa de acceso a los mismos, y los servicios ofrecidos a terceros.

En la capa de recursos podemos encontrar los sensores habilitados para poder recoger información del entorno y el avatar digital del usuario. El avatar digital se compone de un perfil virtual basado en una ontología e incluye herramientas para el manejo de su privacidad e inferencia. Este perfil virtual esta implementado en formato JSON siguiendo la estructura de la ontología diseñada para el mismo.

En la capa de acceso a los recursos encontramos uno de los componentes fundamentales del framework debido a su papel en el apartado de seguridad y privacidad del usuario: el Digital Avatar Contoller (DAC). Este controlador esta implementado mediante una API que supone el único punto de acceso a la información contenida en el avatar digital incluyendo operaciones para el manejo de la información basadas en la ontología y algunas llamadas al sistema, por ejemplo, la que se usa para mostrar notificaciones al usuario. El DAC se encarga de controlar las operaciones que se ejecutan sobre el avatar verificando los permisos y preferencias de privacidad definidas. Dispone de una interfaz sencilla que permite un acceso uniforme tanto desde aplicaciones de propósito específico de terceros instaladas como desde el motor de ejecución que describimos a continuación. Más información acerca del avatar digital y el DAC puede consultarse en nuestro trabajo previo [7].

Por último, encontramos la capa de servicios, que incluye los motores de ejecución y de inferencia. El motor de ejecución es uno de los servicios ofrecidos por el frame-

work para interacciones puntuales y directas con el entorno del usuario. Está implementado usando Beanshell, un intérprete Java que permite la ejecución de scripts directamente en tiempo de ejecución. Típicamente un script puede contener llamadas al DAC para obtener información y adaptar el servicio ofrecido por el dispositivo proveedor del script. El modelo de este servicio de interacción se encuentra descrito en detalle en [10].

El otro servicio de esta capa es el motor de inferencia con tecnología CEP. Este es el sistema en el que se centra este artículo y se describe en la siguiente subsección.

2.2 Mobile Complex Event Processing para detección de ADLs

La tecnología CEP proporciona una arquitectura y herramientas que permiten detectar situaciones complejas mediante el análisis y la correlación de datos provenientes de diferentes fuentes [11]. En el marco de Digital Avatars, el motor CEP integrado en el smartphone extrae información significativa de los datos provenientes de sus propios sensores y las interacciones con smart things mediante la aplicación de una combinación de reglas y patrones CEP. Una de las principales ventajas de CEP es que funciona en tiempo real, sirviendo de soporte para la toma de decisiones con baja latencia.

En nuestra propuesta, usamos el motor Siddhi de WSO2², que ofrece una plataforma extendida con éxito a Android y Raspberry Pi mediante el uso de plugins. El Lenguaje de Procesamiento de Eventos (EPL de sus siglas en inglés) SiddhiQL es muy similar a Esper EPL (que es el EPL más conocido y de referencia basado en SQL para motores CEP) e incluye los principales tipos de operadores necesarios para construir las reglas: seleccionar, filtrar, ventana, patrones, etc. Adicionalmente, el plugin de Android Siddhi IO proporciona al motor capacidades adicionales inherentes a los smartphones, como la gestión de sensores integrados. Combinando estos operadores, podemos construir aplicaciones que monitorizan e infieren el comportamiento y los hábitos de una persona.

El comportamiento del motor Siddhi viene dado por las aplicaciones de propósito específico que se ejecutan en él. De esta forma, terceros pueden desarrollar aplicaciones Siddhi específicas, que se instalan en el motor del smartphone de forma dinámica y permiten a su servicio o aplicación monitorizar el contexto y acciones del usuario y reaccionar frente a determinadas situaciones. Las aplicaciones Siddhi están compuestas por un conjunto de reglas y patrones CEP específicos del dominio junto a declaraciones de *sources* y *sinks*.

Los *sources* son productores de eventos simples para el motor Siddhi. En el complemento de Android Siddhi IO encontramos varios *sources* para cada sensor presente en smartphones. Los eventos generados por un *source* son enviados al motor por diferentes *streams* de datos definidos en las aplicaciones Siddhi. Por otro lado, los *sinks* son los actuadores o *streams* de salida disponibles para producir un efecto cuando el motor detecta un evento de interés. El complemento de Android proporciona *sinks* para emitir sonido, una notificación o un evento sobre el canal Android Broadcast. Este canal es un medio común para elevar eventos del sistema (o de Siddhi en este

² Siddhi CEP engine: <https://siddhi.io/>

caso) que puede ser escuchado por cualquier aplicación instalada en el smartphone siguiendo un modelo de publicación/suscripción. Así, Siddhi permite definir receptores específicos para tratar situaciones de interés detectadas.

Una de las ventajas de Siddhi es que es de código abierto, lo que nos permite extender su funcionalidad agregando nuevos *sources* de datos para sensores no contemplados en el plugin Android. Cualquier sensor independiente que disponga de una interfaz Android es susceptible de ser incluido como un nuevo *source*.

```
from every (e1=pressureSensor[from=='sofa' and value=='1']->
           e2=pressureSensor[from=='sofa' and value=='0'])
select e2.time as time insert into sofa;
from sofa#window.timeBatch(24 hour) select count(time) as count, time
insert into sofaCount;
from sofaCount select count, 'User sat down on sofa {{count}} times
today' as message, time
insert into printCountEventDetected;
```

Código. 1. Extracto de aplicación Siddhi para la monitorización de veces que el usuario se sienta en el sofá como parte de la ADL de movilidad funcional

Supongamos que queremos monitorizar la capacidad del usuario para sentarse y levantarse del sofá de su casa. Esto podría considerarse como parte de la ADL de movilidad funcional del usuario. Solo necesitamos instalar un sensor de presión en el sofá y desarrollar un *source* en Siddhi para este tipo de sensor. Este *source* puede implementarse simplemente extendiendo el tipo *Source* disponible en el plugin Android siguiendo un modelo de interruptor on/off para su comportamiento, de forma que tenga valor 1 cuando se detecta alguna presión y 0 cuando no. Así, una aplicación Siddhi para contar el número de veces que el usuario se ha sentado en el sofá podría ser como se muestra en Código 1. En este ejemplo de aplicación, “pressureSensor”, “sofa”, “sofaCount” y “printCountEventDetected” son *streams* declarados anteriormente en la aplicación, estando el primero asociado al *source* implementado para este tipo de sensor y el último a un *sink* de notificación al usuario.

3 Simulador para AAL

Una de las principales dificultades en el desarrollo de soluciones AAL es la necesidad de la colaboración de usuarios reales para poder probar el sistema antes de implantarlo. La solución a este problema pasa por el uso de simuladores que sean capaces de generar un banco de datos suficientemente realista. Es por esto que, junto al motor CEP de Digital Avatars, proponemos una herramienta para el desarrollo de gemelos digitales de casas inteligentes habitadas, compatible con el framework, con el objetivo de obtener datos sobre las ADLs a través de sensores virtuales.

Los Gemelos Digitales o Digital Twins son réplicas virtuales de entidades físicas [12] con multitud de aplicaciones, por ejemplo, creación de prototipos y rediseño de la entidad física o para predecir su comportamiento. Una de las principales ventajas

de este modelo es la capacidad de trabajar con el gemelo de una manera más flexible, obviando todas las limitaciones físicas de la entidad original.

La herramienta presentada en [13], Simulator of Smart Houses Digital Twins (SiSHoDit), es un simulador completamente automático para ADL que permite comportamientos emergentes en un espacio 3D. La herramienta también admite instancias de incertidumbre en los sensores usando una semilla inicial para modificar pseudoaleatoriamente el resultado de la simulación. El comportamiento de los agentes en SiSHoDit está dirigido por las necesidades de los mismos, como pueden ser el hambre o el sueño. Esto puede observarse en la Figura 1, donde vemos al agente desplazarse del cuarto de baño al dormitorio para reducir su porcentaje de sueño. Estas necesidades se resuelven a través de soluciones compuestas por una serie de pasos en los que el agente interactúa con diferentes objetos, en el caso de estar sensorizados, provocan que se disparen sensores, por ejemplo, un sensor de presión en la cama.



Fig. 1. Visualización del simulador SiSHoDit donde el agente se dispone a dormir para bajar su porcentaje de sueño.

Para el desarrollo del simulador hemos utilizado el motor de juego Godot³. Entre sus características encontramos que es open source, lo que facilitará el acceso a la herramienta por investigadores interesados; y que usa un lenguaje de programación propio pero similar a Python, que tiene además abundante documentación. Las principales ventajas de usar un motor de juego para simulaciones son la inclusión de un editor 3D, que ayuda en la tarea de construir el entorno de forma fácil y visual, además de un motor de renderizado, herramientas para encontrar rutas y un motor físico.

Para el formato de los datos de salida, se eligió JSON, ya que es flexible, ampliamente utilizado y legible, además de ser el lenguaje empleado para almacenar los avatares digitales. El Código 2 muestra un ejemplo de estos archivos generados por el simulador en el que el agente realiza la actividad de ver la televisión. En los datos hay información combinada de las acciones del agente y las salidas del sensor. Cada entrada corresponde a un evento en el simulador y está compuesta por cuatro campos:

³ Juan Linietsky, Ariel Manzur, and Godot Engine community. Godot Engine - Free and open source 2D and 3D game engine: <https://godotengine.org/>

from, que contiene el emisor del evento; *time*, que incluye los segundos transcurridos desde el inicio de la simulación; *type*, que informa sobre el tipo de sensor que generó el evento o el estado de la actividad que está realizando el agente; y *value*, que contiene la salida del sensor o la actividad si el agente es el emisor.

```
[{"from":"person", "time":32769.351034, "type":"activity_begin",
"value":"watch_tv"},
{"from":"televisionModern", "time":32775.684114,
"type":"device", "value":1},
{"from":"loungedesignsofacorner", "time":32782.440996,
"type":"pressure", "value":1},
{"from":"loungedesignsofacorner", "time":36368.18041,
"type":"pressure", "value":0},
{"from":"televisionModern", "time":36372.18025, "type":"device",
"value":0}]
```

Código. 2. Datos de salida del simulador SiSHoDit para la actividad de ver la televisión

Para realizar pruebas del motor CEP de Digital Avatars con los datos que genera el simulador, hemos utilizado AWS Simple Notification Service, una herramienta de publicación/ suscripción con funcionalidad de notificaciones push al smartphone. La forma de uso es muy sencilla, hemos creado canales de suscripción para cada clase de sensor (presión, dispositivo, agua, contacto etc.), así, desde la aplicación de Digital Avatars podemos suscribirnos únicamente a los tipos de sensores que son de interés para nuestra prueba. Las pruebas pueden realizarse a tiempo real, es decir, los datos de la simulación se envían siguiendo los tiempos generados por el simulador, o podemos acelerar la escala de tiempo. Esta funcionalidad es muy interesante, permitiéndonos, por ejemplo, hacer pruebas de una semana de duración en pocos minutos.

Para poder recibir las notificaciones en el motor Siddhi, definimos un nuevo tipo de *source* que hemos llamado “VirtualSensorSource”. Esta clase extiende la funcionalidad genérica de cualquier *source* de Siddhi y añade el receptor de notificaciones. Cuando una notificación llega al “VirtualSensorSource”, se da formato al evento y se clasifica según su tipo de sensor, siendo esta una variable cuando se declara un *source* de este nuevo tipo en la aplicación Siddhi. Si llega una notificación y el tipo del sensor coincide con el especificado en la variable, el evento se envía al *stream* declarado en la aplicación Siddhi para su procesamiento.

4 Trabajos Relacionados

En la literatura podemos encontrar numerosos trabajos relacionados con la monitorización de personas mayores en entornos sensorizados y reconocimiento de ADLs. La mayor parte de estos trabajos están basados en modelos probabilísticos, redes bayesianas, etc. Por ejemplo, [14] usa una red neuronal preentrenada para poder llevar este tipo de aprendizaje al edge computing y que así dispositivos pequeños monitoricen señal de video RGB en la propia casa.

En [15] se propone el uso de una arquitectura CEP para la monitorización de la posición del usuario en exteriores y programación de alarmas de entrada y salida de zonas delimitadas geográficamente. La decisión de usar CEP viene soportada por el hecho de que los entornos de AAL por lo general son ricos en eventos y esta tecnología es capaz de manejar grandes cantidades de eventos de diversas fuentes teniendo además en cuenta la variable del tiempo para su correlación. De esta forma vemos que CEP es también una de las tecnologías más idóneas y con mayor soporte dentro de la comunidad científica para este tipo de sistemas en escenarios de AAL [16].

En relación con el uso de ontología para dotar de estructura semántica a la información almacenada, en [17] proponen un framework basado en la integración de una ontología para información contextual y un motor CEP en un escenario de monitorización de actividad de personas mayores. El motor CEP se utiliza para realizar consultas y análisis temporal de la información almacenada. También en [18] han desarrollado un motor CEP “semántico” propio con una ontología. Se encuentra en desarrollo y hacen pruebas usando el dataset CASAS con información para reconocimiento de ADL. El problema de estos trabajos es que se pierde una de las características más interesantes de CEP que es su funcionamiento para soportar la toma de decisiones en tiempo real en las situaciones de interés detectadas.

En [19] también se propone la utilización de CEP para monitorizar la salud del usuario a través de datos obtenidos de sensores wearables. Sin embargo, en este trabajo se describe un problema relacionado con alojar el motor CEP en un servidor externo al hogar, produciéndose un gran tráfico de datos constante entre los dispositivos que monitorizan signos vitales y el servidor CEP. Así, es necesario establecer una unidad de procesamiento local en el hogar que filtre y agregue los datos, detectando cambios en las señales, enviando la mínima información en tiempo real y reduciendo en gran medida el tráfico de datos. Este es uno de los beneficios que encontramos al implantar el motor CEP directamente en el smartphone.

5 Conclusiones y Trabajos Futuros

En trabajos previos abogamos por el uso del smartphone para mantener y gestionar un avatar digital del usuario que pudiera ser utilizado como base para personalizar las interacciones del usuario con su entorno digital. De esta forma el usuario es dueño de sus datos y decide cómo y con quién compartir su información. Además, la coherencia de este perfil es superior a tener diferentes perfiles para cada uno de los servicios que el usuario utiliza. Así nace Digital Avatars, como un framework que permite a terceros desarrolladores de aplicaciones sociales móviles utilizar servicios basados en la información que quieren obtener del usuario para ofrecerles una experiencia mejor.

En este trabajo mostramos cómo este enfoque es además viable para aplicaciones AAL que necesiten monitorizar la actividad de personas mayores en su hogar para mejorar su independencia. Con la aportación de un motor CEP el framework ahora es capaz de ofrecer un servicio de monitorización con conocimiento del contexto del usuario. Este servicio permite el análisis y correlación de eventos o situaciones detectadas en el tiempo y habilita a aplicaciones de propósito específico para tomar deci-

siones en tiempo real. Adicionalmente y en relación con el caso de estudio, presentamos también un simulador automático basado en gemelos digitales de hogares habitados que permite reducir los costes de tiempo y limitaciones físicas que surgen a la hora de probar sistemas AAL. Este simulador es capaz de generar en poco tiempo datos de sensores implantados en el hogar con gran capacidad de realismo que pueden ser analizados por el framework para probar el entorno diseñado sin necesidad de contar con personas físicas viviendo en él.

Como trabajos futuros nos proponemos mejorar dicho simulador con nuevas funciones como una ampliación en el catálogo de los sensores o mejorar las herramientas para definir el comportamiento de los agentes.

Agradecimientos

Este trabajo ha sido financiado por el Gobierno de España a través del proyecto RTI2018-098780-B-I00, que financia el contrato predoctoral PRE2019-089614, el proyecto TED2021-130296A-I00, por la Junta de Comunidades de Castilla-La Mancha a través del contrato PREJCCM2019/14 y por la beca de movilidad del Vicerrectorado de Profesorado y Desarrollo Profesional de la UCLM.

Referencias

1. Do, T.M., Loke, S.W., Liu, F.: HealthyLife: An Activity Recognition System with Smartphone Using Logic-Based Stream Reasoning. In: Zheng, K., Li, M., and Jiang, H. (eds.) *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. pp. 188–199. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40238-8_16.
2. Murphy, M.J., Peterson, M.J.: Sleep Disturbances in Depression. *Sleep Medicine Clinics*. 10, 17–23 (2015). <https://doi.org/10.1016/j.jsmc.2014.11.009>.
3. Saeb, S., Zhang, M., Karr, C.J., Schueller, S.M., Corden, M.E., Kording, K.P., Mohr, D.C.: Mobile Phone Sensor Correlates of Depressive Symptom Severity in Daily-Life Behavior: An Exploratory Study. *Journal of Medical Internet Research*. 17, e4273 (2015). <https://doi.org/10.2196/jmir.4273>.
4. Aarts, E., Marzano, S.: *The New Everyday: Views on Ambient Intelligence*. 010 Publishers (2003).
5. How far we've come: a look at smartphone performance over the past 7 years, <https://www.androidauthority.com/smartphone-performance-improvements-timeline-626109/>, last accessed 2023/03/09.
6. Stipkovic, S., Bruns, R., Dunkel, J.: Pervasive Computing by Mobile Complex Event Processing. In: 2013 IEEE 10th International Conference on e-Business Engineering. pp. 318–323 (2013). <https://doi.org/10.1109/ICEBE.2013.49>.
7. Perez-Vereda, A., Hervas, R., Canal, C.: Digital Avatars: A programming framework for personalized human interactions through virtual profiles. *Pervasive and Mobile Computing*. 87, 101718 (2022). <https://doi.org/10.1016/j.pmcj.2022.101718>.
8. Bertoa, M.F., Moreno, N., Perez-Vereda, A., Bandera, D., Álvarez-Palomo, J.M., Canal, C.: Digital Avatars: Promoting Independent Living for Older Adults. *Wireless Communi-*

- cations and Mobile Computing. 2020, e8891002 (2020). <https://doi.org/10.1155/2020/8891002>.
9. Guillén, J., Miranda, J., Berrocal, J., García-Alonso, J., Murillo, J.M., Canal, C.: People as a Service: A Mobile-centric Model for Providing Collective Sociological Profiles. *IEEE Software*. 31, 48–53 (2014). <https://doi.org/10.1109/MS.2013.140>.
 10. Pérez-Vereda, A., Canal, C., Pimentel, E.: Modelling digital avatars: A tuple space approach. *Science of Computer Programming*. 203, 102583 (2021). <https://doi.org/10.1016/j.scico.2020.102583>.
 11. Luckham, D.C.: *Event Processing for Business: Organizing the Real-Time Enterprise*. John Wiley & Sons (2011).
 12. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*. 29, 36–52 (2020). <https://doi.org/10.1016/j.cirpj.2020.02.002>.
 13. Cabañero, L., Perez-Vereda, A., Nugent, C., Cleland, I., Hervas, R., González, I.: A Software Tool and a Metamodel for Digital Twins of Inhabited Smart Environments. In: Bravo, J., Ochoa, S., and Favela, J. (eds.) *Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022)*. pp. 747–759. Springer International Publishing, Cham (2023). https://doi.org/10.1007/978-3-031-21333-5_74.
 14. Sufian, A., You, C., Dong, M.: A deep transfer learning-based edge computing method for home health monitoring. Presented at the 2021 55th Annual Conference on Information Sciences and Systems, CISS 2021 (2021). <https://doi.org/10.1109/CISS50987.2021.9400321>.
 15. Târnaucă, B., Puiu, D., Nechifor, S., Comnac, V.: Using complex event processing for implementing a geofencing service. Presented at the SISY 2013 - IEEE 11th International Symposium on Intelligent Systems and Informatics, Proceedings (2013). <https://doi.org/10.1109/SISY.2013.6662608>.
 16. Shahad, R.A., Saad, M.H.M., Hussain, A.: Activity recognition for smart building application using complex event processing approach. *International Journal on Advanced Science, Engineering and Information Technology*. 8, 315–322 (2018). <https://doi.org/10.18517/ijaseit.8.2.2575>.
 17. Culmone, R., Falcioni, M., Giuliadori, P., Merelli, E., Orrù, A., Quadrini, M., Ciampolini, P., Grossi, F., Matrella, G.: AAL domain ontology for event-based human activity recognition. Presented at the MESA 2014 - 10th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Conference Proceedings (2014). <https://doi.org/10.1109/MESA.2014.6935631>.
 18. Trăscău, M., Sorici, A., Florea, A.: Detecting activities of daily living using the CONSERT engine. In: *Advances in Intelligent Systems and Computing*. pp. 94–102 (2019). https://doi.org/10.1007/978-3-030-01746-0_11.
 19. Romano, L., Coppolino, L., Elia, I.A., Spagnuolo, G.: A healthcare real-time monitoring system for multiple sensors data collection and correlation. In: Damiani, E., Jeong, J., Howlett, R.J., and Jain, L.C. (eds.) *New Directions in Intelligent Interactive Multimedia Systems and Services - 2*. pp. 455–464. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02937-0_42.