Análisis inteligente de flujos de trabajo sociales *

Manuel Lama[1], Pedro Álvarez[2], Manuel Ocaña[3], Manuel Mucientes[1], Joaquín Ezpeleta[2], Miguel Ángel Garrido[3], Alberto Bugarín[1]

[1] Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS)
Universidade de Santiago de Compostela, Santiago de Compostela, España
{manuel.lama, manuel.mucientes,alberto.bugarin.diz}@usc.es
[2] Instituto de Investigación en Ingeniería de Aragón (I3A)
Depto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, España
{alvaper,ezpeleta}@unizar.es
[3] Depto. de Electrónica, Escuela Politécnica Superior.
Universidad de Alcalá de Henares, Alcalá de Henares, España.
{mocana,garrido}@depeca.uah.es

Abstract. Los flujos de trabajo sociales (SOW) coordinan las actividades realizadas por un conjunto de usuarios que bien de forma individual o en cooperación tratan de alcanzar un determinado objetivo. Los SOW son flujos no estructurados en los que participan un gran número de usuarios que llevan a cabo actividades de muy diversa naturaleza que se extienden a lo largo del tiempo y que típicamente consumen pocos recursos de computación. Un ejemplo de procesos que se modelan a través de este tipo de flujos de trabajo son las campañas de marketing que tienen como objetivo motivar a los potenciales clientes en el consumo de un determinado producto o servicio. En este trabajo, se presenta el proyecto *Inteligencia (Artificial) de Negocio para Flujos de Trabajo Sociales*, en el que se combinan técnicas de minería de procesos, estrategias de paralelización de algoritmos, y técnicas de localización y seguimiento de usuarios, con el fin de extraer información relevante sobre SOW, como los que modelan, entre otros, el comportamiento de los usuarios en campañas de marketing desarrolladas en escenarios abiertos.

1 Introdución

Los *flujos de trabajo sociales* (SOW) modelan la coordinación de la ejecución de una serie de actividades que deben ser realizadas por un conjunto de usuarios [7]. Estos usuarios pueden actuar de forma individual o colaborativa para alcanzar un determinado objetivo, potenciando así el carácter social del flujo de trabajo. Ejemplos de este tipo de flujos de trabajo (WF) son las campañas de marketing que se realizan en espacios abiertos o la formación de empleados a través de estrategias de *gamificación*. No obstante, desde un punto de vista tecnológico las

* Los autores desean agradecer al Ministerio de Economía y Competitividad el soporte financiero a través de los proyectos TIN2014-56633-C3-1-R, TIN2014-56633-C3-2-R, TIN2014-56633-C3-3-R y TIN2014-53986-REDT.

principales características de los SOW son: *(i)* modelan *procesos no estructurados* en los que los usuarios disponen de múltiples opciones y/o caminos y donde incluso pueden realizar actividades adicionales a las inicialmente previstas en el flujo de trabajo; *(ii)* pueden participar simultáneamente un elevado número de usuarios; y *(iii)* en función de la habilidad o motivación de cada usuario, una misma actividad puede ser realizada de inmediato o tener una duración temporal mucho mayor (del orden de horas o incluso días).

Estas características añaden una serie de retos a la hora de gestionar y monitorizar la ejecución de los SOW. Por una parte, la gestión de este tipo de WF requiere el manejo de un gran número de ejecuciones simultáneas cuya duración se extiende a lo largo del tiempo. Por otra parte, el análisis de la ejecución de los WF requiere del uso de técnicas de minería de procesos [1] para obtener el WF *real* seguido por los usuarios y, de este modo, *entender* qué es lo que realmente ha pasado en la ejecución del proceso, permitiendo con ello la mejora del proceso y su adaptación (dinámica) a las necesidades de los usuarios. Sin embargo, para procesos no estructurados con un número enorme de ejecuciones, los resultados de los algoritmos actuales de descubrimiento de procesos son WF que tienen una estructura en *espagueti* y que, por tanto, son muy difíciles, cuando no imposibles, de interpretar. Por ello, es necesario desarrollar técnicas que permitan extraer información valiosa sobre el flujo de trabajo descubierto. Este es el primer eje alrededor del cual pivota el proyecto *Inteligencia (Artificial) de Negocio para Flujos de Trabajo Sociales* (BAI4SOW).

Por otra parte, el proyecto BAI4SOW añade otra dimensión a los SOW: las actividades a realizar por parte de los usuarios tienen lugar en escenarios o áreas geográficas de exteriores (como puede ser un centro comercial abierto, una zona de interés turístico, un área donde se desarrolle alguna actividad o, en general, cualquier zona de una ciudad) y además son actividades geoposicionadas que los usuarios llevan a cabo a través de dispositivos móviles (como las actividades de una campaña de marketing). Por tanto, es necesario capturar los eventos relacionados con la ejecución de este tipo de actividades y, para ello, se requiere el desarrollo de técnicas de localización y de detección del comportamiento de los usuarios en un área geográfica dada. Este es el segundo eje del proyecto BAI4SOW.

En este artículo, se presenta el proyecto BAI4SOW cuyo objetivo consiste en el desarrollo de algoritmos de minería de procesos para el análisis de SOW que contienen actividades geoposicionadas. Además, dado el elevado número de usuarios que potencialmente pueden participar en este tipo de flujos es necesario ejecutar los algoritmos en infraestructuras de cómputo *grid*, *clúster* y *cloud*, minimizando el coste que podría tener dicho cómputo en *clouds* empresariales como Amazon y Google. Este es el tercer eje del proyecto BAI4SOW.

El artículo se estructura de la siguiente manera: en la Sección 2 se presenta el marco conceptual del proyecto; en las Secciones 3, 4 y 5 se describe cómo se localiza a los usuarios, los algoritmos de minería de procesos y la capa *middleware* que gestiona las infraestructuras de cómputo, respectivamente. Finalmente, la Sección 6 presenta las conclusiones del trabajo y el trabajo futuro.

2 Marco conceptual

BAI4SOW tiene como objetivo general el desarrollo de técnicas inteligentes para la extracción automática y el análisis del comportamiento de los usuarios en procesos modelados a través de flujos de trabajo sociales, en general, y en particular, en aquellos procesos que involucran a usuarios que se mueven en un área geográfica exterior y que ejecutan las actividades del flujo social a través de sus dispositivos móviles. Para alcanzar este objetivo se ha propuesto un marco conceptual que consta de los siguientes componentes (Figura 1):

- *Algoritmos de localización y detección de comportamientos.* Para analizar los SOW es necesario registrar el conjunto de eventos generados por las actividades que realiza cada usuario en una ejecución de dichos flujos, de modo que para cada ejecución se registra una traza de estos eventos. En procesos que involucran el desplazamiento de los usuarios en un área geográfica, estas actividades consisten en llegar a un punto determinado (para, por ejemplo, realizar una compra o tomar una foto), sentarse o moverse dentro de un local, etc. Por tanto, es necesario *localizar* a los usuarios en el área geográfica sobre la que se mueven, así como detectar sus comportamientos de interés una vez se encuentran en una localización dada. Para registrar la realización de estas actividades se desarrollan una serie de algoritmos que hacen uso del GPS, de la señal WiFi y de los sensores inerciales de los dispositivos móviles de los usuarios.
- *Algoritmos de minería de procesos.* Este conjunto de algoritmos permiten, por una parte, *descubrir automáticamente* el WF *real* que ha sido ejecutado por los usuarios y, por otra parte, extraer información relevante sobre dicho WF, como los *patrones frecuentes* de actividades, que indican modelos

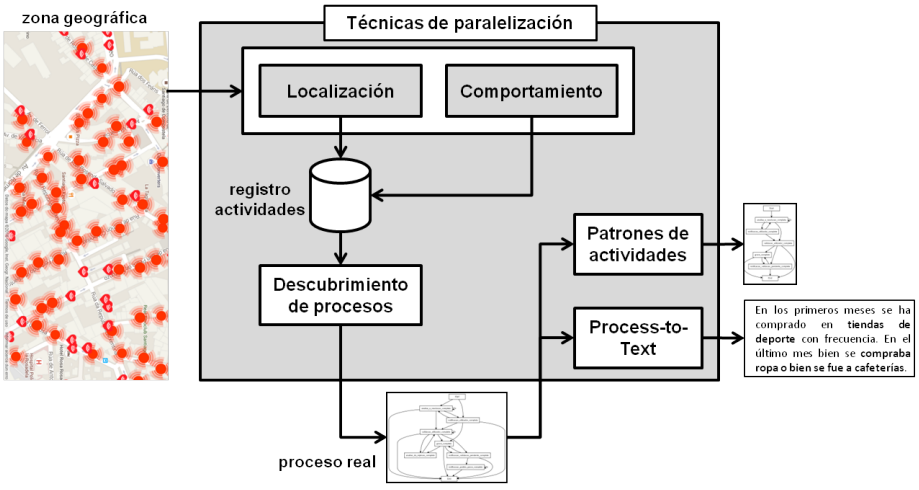

Fig. 1. Marco conceptual del proyecto.

de comportamiento frecuente por parte de los usuarios; la *jerarquización* de WF no estructurados y en *espagueti*, que ayuda a visualizar el proceso al reducir su complejidad en diferentes niveles de abstracción; y la *descripción lingüística* de dichos WF, con la que se relatan de forma automática y en lenguaje natural sus características más relevantes, proporcionando una información complementaria a la que obtienen los usuarios mediante la visualización a través de grafos.
- *Técnicas de paralelización de algoritmos.* Se hará uso de este tipo de técnicas para diseñar una infraestructura tecnológica basada en la integración de recursos *grid* y *cloud* que ofrezca soporte para el procesamiento eficiente y a gran escala de los algoritmos desarrollados en el marco del proyecto. Esta infraestructura integrará modelos de coste que faciliten la decisión de qué recursos computacionales y de almacenamiento son más apropiados en cada instante para la ejecución de los algoritmos, contemplando la posibilidad de utilizar recursos de diferentes infraestructuras o proveedores.

Es importante resaltar que este marco conceptual es aplicable a SOW que contienen actividades relacionadas con la localización y el comportamiento de los usuarios que participan en ellos. No obstante, las técnicas de minería de procesos y de paralelización de algoritmos son directamente aplicables a cualquier WF, en general, y en particular a flujos no estructurados y en los que participan un enorme número de usuarios. En las siguientes secciones se detallan cada uno de los componentes del marco conceptual.

3 Localización y detección de comportamientos

El objetivo de este componente del marco conceptual es doble: por una parte, desarrollar un *sistema de localización* de usuarios en zonas de interés tanto para interiores como exteriores, y por otra parte, identificar y reconocer de forma automática el conjunto de actividades que definen el comportamiento de un usuario cuando se encuentra en un punto geográfico dado (por ejemplo, un local comercial). Así:

- El proceso de localización está basado en un sistema jerárquico que proporciona la estimación de la posición del usuario en una zona geográfica dada (como un centro comercial), empleando para ello: *(i)* un sistema de localización de alto nivel que identifica las zonas o puntos de interés por medio del empleo del GPS en exteriores y el sistema de localización con *fingerprints* propuesto por los autores en [9] enriquecido con *(ii)* un sistema de localización de bajo nivel que realiza el seguimiento entre posiciones o zonas de interés, haciendo una fusión multisensorial basada en Filtro Extendido de Kalman (EKF) con los sensores inerciales del dispositivo móvil (giróscopos, acelerómetros y podómetros), como se muestra en la Figura 2.
  Para mejorar el cálculo de la orientación se suele recurrir al uso de filtros, como el filtro de Mahony [11], que es un filtro complementario eficiente para sensores inerciales, aunque tiene un error constante en la orientación, al no

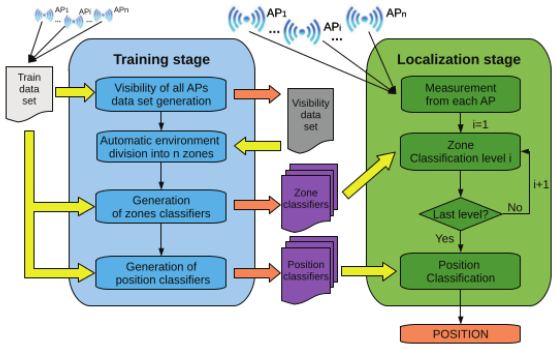

Fig. 2. Localizador de alto nivel.

disponer de un observador absoluto ortogonal a la gravedad. Para resolver este problema se puede usar el filtro de Madgwick [10], como un filtro de gradiente descendiente optimizado que calcula la orientación incluyendo la compensación de la distorsión magnética y la compensación del bias del giróscopo. De esta manera, añade al filtro de Mahony las medidas del magnetómetro y, por tanto, se obtiene la orientación en las tres dimensiones. Así, para obtener la orientación de la unidad de medida inercial (IMU) del dispositivo móvil (Figura 3), en primer lugar se calibrará y, a continuación, se usará el filtro de Madgwick para compensar el *bias* del giróscopo. Posteriormente, se fusionan estos datos con el podómetro del móvil mediante el filtro EKF.

– La detección del comportamiento de los usuarios exige describir cualitativamente el conjunto de comportamientos (o actividades) que se desea detectar. Para ello se usa un modelo lingüístico jerárquico con varios niveles de abstracción que describe de forma incremental las actividades de los usuarios,

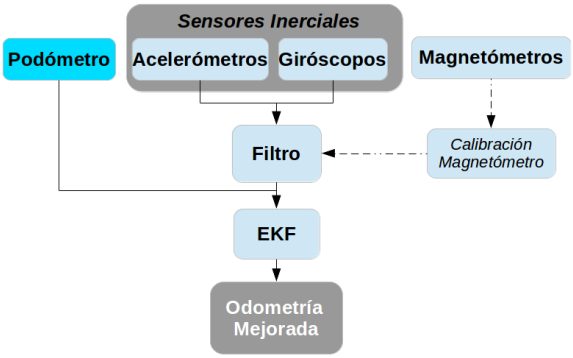

Fig. 3. Sistema de seguimiento propuesto.

estableciendo hipótesis a partir de la información de eventos de más bajo nivel proporcionados por los sensores disponibles en el dispositivo móvil. El paso de datos sensoriales de bajo nivel a información cualitativa de alto nivel se realizará mediante un algoritmo inteligente de fusión multisensorial basado en técnicas de *soft computing* adaptadas a entornos de procesamiento masivo que permiten manejar los datos de miles de usuarios simultáneamente.

Como resultado, este componente genera y registra los eventos asociados con la ejecución de las actividades de localización y de comportamiento de usuarios que forman parte de los SOW. Estos eventos contienen información sobre cuándo y dónde ha tenido lugar la actividad, así como quién la ha realizado. Esta información será la que usarán los algoritmos de minería de procesos para obtener los WF *reales* que han sido seguido por los usuarios.

4 Algoritmos de minería de procesos

Las técnicas de minería de procesos [1] permiten descubrir automáticamente, y a partir de ficheros de registro, el WF *real*, o comportamiento, que han seguido los usuarios que participan en un SOW. Estos ficheros de registro presentan dos características que condicionan enormemente el tipo de algoritmos a desarrollar: la gran cantidad de usuarios que potencialmente pueden participar en un SOW (del orden de decenas de miles) y el elevado número de eventos asociados a actividades que puede generar cada usuario (decenas de actividades). En estos casos, los WF descubiertos por los algoritmos tienen una estructura en *espagueti* que dificulta su comprensibilidad, ya que existe un enorme número de relaciones causales entre las actividades que conforman el WF. Para facilitar esta comprensibilidad por parte de los gestores del WF, se aplicarán las siguientes técnicas de minería de procesos:

- *Algoritmo de descubrimiento de WF.* Este algoritmo debe generar WF que sean completos, precisos y simples, considerando los elevados niveles de ruido que contienen los registros de eventos de actividades en los SOW. Así, el algoritmo debe maximizar los criterios de completitud, precisión y simplicidad, ya que pueden existir caminos de control poco frecuentes pero que son muy informativos, filtrando en la medida de lo posible el ruido que añaden los usuarios al completar actividades no informativas o no completar los WF. Uno de los algoritmos que tiene en cuenta estos requisitos es ProDiGen [13], un algoritmo evolutivo basado en una función jerárquica de evaluación que prioriza la completitud frente a la precisión y ésta frente a la simplicidad. Este algoritmo está integrado en la plataforma ProDiGen que se está desarrollando en el marco del proyecto (Figura 4).
- *Algoritmo de identificación de patrones frecuentes.* A través de este algoritmo, y partiendo del fichero de registro y del WF descubierto, se identifican los patrones de actividades que los usuarios realizan con *frecuencia* durante la ejecución del WF. De esta forma, se describen únicamente aquellos patrones que son ejecutados con una frecuencia superior a un umbral dado,

Fig. 4. Herramienta ProDiGen de minería de procesos.

lo que permite entender el comportamiento de los usuarios. Para ello se ha desarrollado un algoritmo basado en el algoritmo *w-find* [8] que ha sido extendido para dar soporte a la identificación de patrones de actividades con bucles de cualquier longitud [4]. Este algoritmo también está integrado en la plataforma ProDiGen.

– *Algoritmo de jerarquización de WF.* El objetivo de este algoritmo consiste en proporcionar diferentes niveles de abstracción de los WF descubiertos [3], obteniendo representaciones más simples en cada uno de estos niveles y facilitando con ello la legibilidad del WF. El algoritmo de jerarquización hace uso de técnicas de agrupamiento de trazas en las que los grupos de trazas se crean en función de su similaridad.

Además de este conjunto de algoritmos basados en técnicas de minería de procesos con los que se pretende describir visualmente el comportamiento de los usuarios, se desarrollará una herramienta [12] orientada a la *descripción lingüística* de los WF reales que han sido obtenidos por el algoritmo de descubrimiento. Esta herramienta actuará como un generador automático de informes en lenguaje natural a partir del proceso descubierto (herramienta *process-to-text*), proporcionando información relevante y seleccionada sobre aquellos aspectos de interés del WF, tales como los patrones y caminos frecuentes, las actividades realizadas un mayor número de veces en un período dado, o la frecuencia de las relaciones entre actividades. Es importante resaltar que estas descripciones textuales son complementarias a la representación visual de los WF a través de grafos de actividades.

5 Paralelización de algoritmos

En el proyecto BAI4SOW, los SOW modelan procesos en los que potencialmente pueden participar miles o decenas de miles de usuarios, como puede ser las campañas de marketing en centros comerciales o los procesos de formación en grandes compañías. Esta característica conlleva la necesidad de paralelizar los algoritmos desarrollados y de ejecutar estas versiones paralelas en infraestructuras de cómputo *grid*, *clúster* y *cloud*.

El punto de partida es una capa software (a nivel de *middleware*) capaz de integrar una serie de infraestructuras de cómputo *clúster* y *grid* que tenemos a nuestra disposición [5]. Esta capa ofrece la funcionalidad típica de un *framework de gestión de recursos*. Todos los recursos de estas infraestructuras son vistos como un único y potente recurso de cómputo y almacenamiento sobre el que es posible ejecutar las versiones paralelas de los algoritmos de minería de procesos y de localización y detección del comportamiento de los usuarios. La Figura 5 muestra el diseño arquitectural de esta capa intermediaria. Su arquitectura está basada en un bus de mensajes que coordina la actividad de los componentes que soportan el ciclo de vida de la ejecución de una aplicación intensa en cómputo y datos. Entre los componentes existentes están los responsables de la gestión y provisión de los recursos, el *scheduling* de las tareas a ejecutar, la gestión y recuperación de fallos, el movimiento de datos entre recursos, la

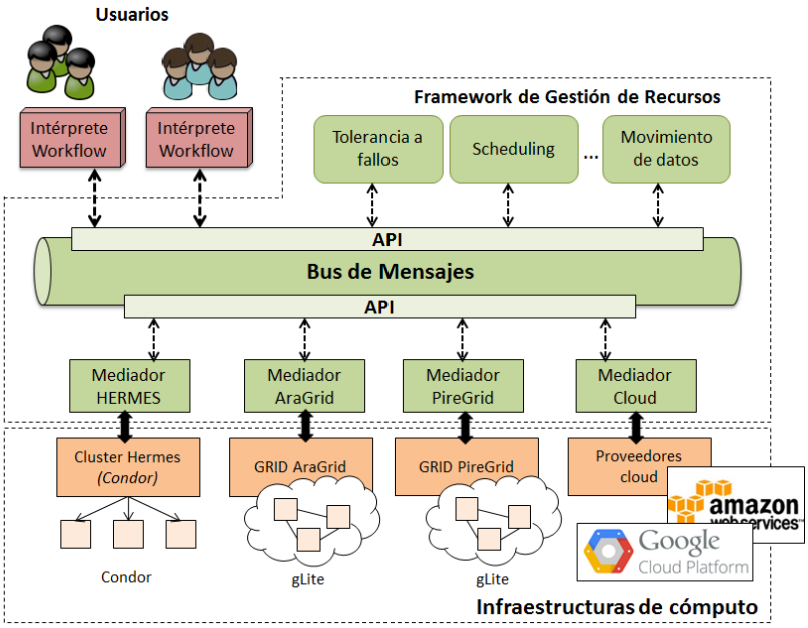

Fig. 5. Arquitectura de alto nivel de la capa *middleware*

monitorización del estado de ejecución, y los mediadores que facilitan el acceso a las infraestructuras de cómputo que fueron integradas. Esta capa software ha sido anteriormente utilizada para resolver con éxito problemas que presentaban fuertes requisitos de cómputo y datos, por ejemplo, la implementación paralela del sistema de anotación semántica presentado en [6].

En BAI4SOW se pretenden potenciar las capacidades de la capa *middleware* para su uso y explotación en entornos de empresa. En la mayoría de los casos, las infraestructuras de cómputo disponibles en el contexto académico (*clúster* o *grid*) no están pensadas para dar soporte a la ejecución de sistemas en producción: sus recursos sólo pueden ser usados para fines relacionados con la investigación, o su fiabilidad, disponibilidad y rendimiento no son adecuados. Para resolver este problema, la funcionalidad de la capa *middleware* será extendida para integrar y gestionar eficientemente los recursos y servicios ofrecidos por los actuales proveedores de *cloud* a nivel de *Infrastructure-as-a-Service* (Amazon AWS, Google Cloud Platform o Microsoft Azure, entre otros). Un primer prototipo de esta nueva versión del *middleware* ya ha sido implementado y probado en el entorno de Amazon AWS. Este prototipo a día de hoy permite ejecutar sobre recursos *cloud* los distintos algoritmos de minería de procesos y de localización y detección del comportamiento de los usuarios, abriendo de esta manera la posibilidad de explotar nuestros resultados en entornos reales de empresa. No obstante, esta evolución hacia entornos tipo *cloud* ha abierto una nueva línea de trabajo en la que abordan cuestiones críticas como, por ejemplo, ¿cómo minimizar el coste de ejecutar estas aplicaciones en una infraestructura donde se paga por el uso de los recursos?, ¿qué recursos son los más adecuados para ejecutar una aplicación o parte de una aplicación?, ¿dónde y cómo almacenar los datos teniendo en cuenta los recursos que han sido contratados?, etc. Todas estas cuestiones se enmarcan en lo que se ha denominado la *economía del cloud*.

En [2] definimos un primer intento de responder a las cuestiones anteriores para el caso de ejecutar aplicaciones del tipo *bag-of-tasks* sobre Amazon AWS. Un modelo de minimización de costes aplicado sobre el catálogo de recursos del proveedor pretendía determinar la mejor combinación de recursos para ejecutar una aplicación teniendo en cuenta los requisitos del usuario (presupuesto o tiempo máximo de ejecución, por ejemplo) y el rendimiento de las instancias contratadas. En BAI4SOW será necesario definir nuevos modelos de minimización que se adecuen a los algoritmos involucrados e integrar estos modelos en el ciclo de vida de la capa *middleware* previamente desarrollada. El objetivo final debe ser que el aprovisionamiento de los recursos cloud sea bajo criterios de coste y rendimiento, y el *scheduling* de las aplicaciones sobre estos recursos sea automático desde la perspectiva del usuario final.

6 Conclusiones

En el proyecto BAI4SOW se combinan de forma exitosa técnicas de minería de procesos, estrategias de paralelización de algoritmos, y técnicas de localización y seguimiento de usuarios para extraer información valiosa sobre la ejecución de

SOW en los que los usuarios realizan actividades en escenarios abiertos como centros comerciales abiertos.

Actualmente, el proyecto se encuentra en una fase de paralelización de los algoritmos que ya han sido desarrollados para que puedan ser desplegados en las infraestructuras de cómputo *grid* y *clúster*. Una vez se haya completado ese despliegue se realizarán una serie de pilotos en entornos con usuarios reales.

References

1. van der Aalst, W.M.P.: Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
2. Álvarez, P., Hernández, S., Fabra, J., Ezpeleta, J.: Cost estimation for the provisioning of computing resources to execute bag-of-tasks applications in the amazon cloud. In: 12th International Conference on Economics of Grids, Clouds, Systems and Services (GECON) 2015. Cluj-Napoca, Romania (2015)
3. Bose, R.P.J.C., Verbeek, H.M.W.E., van der Aalst, W.M.P.: Discovering hierarchical process models using prom. In: Nurcan, S. (ed.) Proceedings of the CAiSE Forum 2011. CEUR Workshop Proceedings, vol. 734, pp. 33–40. London, UK (2011)
4. Chapela, D., Vazquez-Barreiros, B., Mucientes, M., Lama, M.: Mining frequent pattern in workflows. In: International Workshop on Algorithms & Theories for the Analysis of Event Data (ATAED 2016) (2016), (Submited)
5. Fabra, J., Álvarez, P., Bañares, J., Ezpeleta, J.: DENEB: a platform for the development and execution of interoperable dynamic web processes. Concurrency and Computation: Practice and Experience 23(18), 2421–2451 (2011)
6. Fabra, J., Hernández, S., Otero, E., Vidal, J.C., Lama, M., Álvarez, P.: Integration of grid, cluster and cloud resources to semantically annotate a large-sized repository of learning objects. Concurrency and Computation: Practice and Experience 27(17), 4603–4629 (2015)
7. Gorg, S., Bergmann, R.: Social workflows - Vision and potential study. Information Systems 50, 1–19 (2015)
8. Greco, G., Guzzo, A., Manco, G., Saccà, D.: Mining and reasoning on workflows. IEEE Transactions on Knowledge & Data Engineering 17(4), 519–534 (2005)
9. Hernández, N., Alonso, J.M., Ocaña, M.: Hierarchical approach to enhancing topology-based WiFi indoor localization in large environments. Journal of Multiple-Valued Logic and Soft Computing 3(5), 221–241 (2016)
10. Madgwick, S.O.: An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK) (2010)
11. Mahony, R., Hamel, T., Pflimlin, J.M.: Nonlinear complementary filters on the special orthogonal group. IEEE Transactions on Automatic Control 53(5), 1203–1218 (2008)
12. Ramos-Soto, A., Bugarín, A., Barro, S.: On the role of linguistic descriptions of data in the building of natural language generation systems. Fuzzy Sets and Systems 285, 31–51 (2016)
13. Vázquez-Barreiros, B., Mucientes, M., Lama, M.: ProDiGen: Mining complete, precise and minimal structure process models with a genetic algorithm. Information Sciences 294, 315–333 (2015)