

Identifying Requirements for Implementing ad-hoc Geo-references BPM

P. Antunes, G. Zurita , N. Baloian

Abstract— There are many scenarios in which business processes will benefit from the integration of geographical information for its management. In this paper we discuss a set of requirements for ad-hoc geo-referenced Business Process Management (BPM), noting in particular the conflicts between spatial and task dependencies when coordinating activities. We suggest the predominance of spatial dependencies and propose the integration of process models in georeferencing tools. We analyse the communication needs of geo-referenced and BPM processes and suggest the adoption of microblogging platforms for coordination support. We also discuss the implementation of an ad-hoc georeferenced BPM tool, specify the microblogging messages needed to coordinate georeferenced activities, and discuss a preliminary formative evaluation of the proposed implementation.

Keywords— BPM, GIS, geo-referenced processes, ad-hoc BPM.

I. INTRODUCCIÓN

Muchos negocios requieren de la gestión integrada de procesos de negocios e información geográfica o espacial. Por ejemplo, en Nueva Zelanda es común contratar servicios privados de compañías de recojo de basura, de modo que cada semana bajo un horario y ruta determinados, un camión viene a recoger los contenedores de basura. Regularmente estos contratos son cancelados y otros nuevos son realizados, que implican cambios y optimización continuos en la ruta. Los sistemas de Información Geográfica o GIS (Geographical Information Systems), ayudan a definir y optimizar visualmente estas rutas. Sin embargo, la definición de una ruta puede ser vista como un proceso de negocio ad-hoc, compuesta de un conjunto de tareas a realizar [1]. El término ad-hoc se refiere a la naturaleza cambiante de los procesos de negocios, que en este caso son constantes y en tiempo real.

Por tanto, se propone un sistema que integra las funcionalidades provistas por sistemas GIS y BPM, como soporte al escenario ya descrito y otros similares, tales como manutención de infraestructura urbana, procesos de lucha contra incendios, reparación de alumbrado público, rondas policiales, etc. Para lo anterior, se ha analizado los requerimientos de coordinación de actividades colaborativas que requieren una gestión ad-hoc BPM geo-referenciada, enfatizando en los conflictos en la coordinación de actividades, producidas por las dependencias de las tareas o bien las espaciales. Nuestro análisis, sugiere la predominancia de las dependencias espaciales, por sobre las tareas;

y por tanto proponemos la integración de modelos de procesos con herramientas de geo-referenciación. Asimismo, analizamos los requerimientos de comunicación de los procesos integrados de GIS y ad-hoc BPM, proponiendo la adopción de una plataforma de comunicación tipo microblogging para dar soporte a la coordinación de los usuarios.

En la sección 2 se discuten los requerimientos de la aplicación ad-hoc BPM geo-referenciada, enfocándose en los conflictos entre las dependencias espaciales y el de las tareas. En la sección 3, se analiza la literatura relacionada. En la sección 4 se discute acerca de la implementación de la herramienta ad-hoc BPM geo-referenciada y se especifica formalmente el set de mensajes del tipo microblogging necesarios para coordinar. Finalmente, en la sección 5, se presentan los resultados preliminares de la evaluación de la herramienta.

II. REQUERIMIENTOS

BPM presupone la existencia de dos aspectos fundamentales: Process Aware Information Systems (PAIS) y modelos de procesos [2]. El acrónimo PAIS no se refiere a un sistema en particular, sino a una categoría de sistemas que adoptan una vista de procesos donde los objetivos de los negocios son descompuestos en un número discreto de actividades. Esta vista de procesos también introduce la noción de coordinación a través de las dependencias de las tareas; es decir, las actividades son coordinadas a través de las reglas de precedencia mediante patrones de workflow [3]. Los modelos de procesos especifican las tareas y las dependencias que son prototipables para un proceso de negocio en particular. Estos modelos descomponen la especificación de procesos desde su especificación, y permiten implementar sistemas basados en modelos de alto nivel que son más fáciles de especificar que códigos de software de bajo nivel.

A su vez, un GIS presupone dos componentes relevantes: la transformación de datos y la visualización [4]. La transformación de datos consiste en la adquisición y presentación de datos geográficos en el computador, mientras que la visualización soporta el razonamiento espacial y la toma de decisiones. Un GIS induce dependencias espaciales, es decir que las actividades de trabajo tienden a estar centradas en los lugares geográficos o regiones, y la coordinación es está implícitamente orientada a cambiar la atención al enfoque espacial. Los usuarios realizan alguna actividad de razonamiento relacionado con cierta región, y cuando ellos se “mueven” a otra, implícitamente empiezan una actividad diferente.

La dicotomía entre las dependencias espaciales y la de

P. Antunes, University of Wellington, Wellington, Nueva Zelanda, pedro.antunes@vuw.ac.nz

G. Zurita, Universidad de Chile, Diagonal Paraguay 257, Santiago, Chile, gzurita@fen.uchile.cl

N. Baloian, Universidad de Chile, Beauchef 850, Santiago, Chile, nbaloian@dcc.uchile.cl

tareas, nos lleva a la interrogante de cómo el trabajo debiera ser coordinado en un sistema ad-hoc BPM geo-referenciado. El proceso de recolección de basura ejemplificado antes, puede ser modelado ya sea mediante la selección de un conjunto de lugares asociando tareas a realizar, o bien puede ser modelado como un conjunto de tareas que se ejecutan consecutivamente, cada una en un lugar diferente. Ambos modelos conducen al mismo resultado, aunque en este ejemplo la primera opción es la más adecuada, ya que las actividades están más relacionadas con la contingencia de relaciones espaciales. En efecto, nosotros hemos experimentado con herramientas geo-colaborativas en varios escenarios [5,6], y se ha observado que en la mayoría de estos, la coordinación se centra en los espacios o lugares, y no en las tareas; es decir, las dependencias de las tareas son secundarias a las dependencias espaciales. Lo anterior nos conduce a indicar el primero de nuestros requerimientos:

R1. En un sistema ad-hoc BPM geo-referenciado, la coordinación debiera ser determinada primordialmente por las dependencias espaciales y solamente de manera secundaria por las dependencias de las tareas.

Esto provee la flexibilidad de que al menos dos opciones pueden ser consideradas respecto a la estructura espacial de los datos: 1) mediante un modelo espacial que define un camino entre distintos lugares; y por lo tanto, exista una secuencia de puntos a recorrer; o 2) no exista tal camino y los usuarios pueden seleccionar la región más conveniente para trabajar. En cualquier caso, se puede ver el proceso de negocio como una colección de lugares o regiones, cada uno teniendo actividades asociadas. Los GIS usualmente no imponen muchas dependencias de tareas y por lo tanto, esto nos indica que un sistema ad-hoc BPM geo-referenciado no debiera tampoco imponer restricciones en como los usuarios interactúan con los elementos espaciales. Este razonamiento nos conduce a indicar los siguientes dos requerimientos:

R2. Un sistema ad-hoc BPM geo-referenciado debería ser considerado como un proceso ad-hoc donde los usuarios determinar el orden de las actividades y los lugares/regiones proveen el contexto.

R3. Cada lugar/región debería tener asociado sub-procesos.

R3 acomoda la combinación de razonamiento espacial y workflow ad-hoc, y puede ser modelado con lenguajes de modelamiento de procesos. Por ejemplo, en BPMN (Business Process Management Notation), esto significa tener un proceso padre ad-hoc con varios sub-procesos, uno para cada lugar/región, y donde los procesos asociados corresponden a un conjunto de actividades y dependencias relacionadas a tal lugar/región.

Habiendo sugerido que un proceso geo-referenciado debería ser un proceso ad-hoc, aun no hemos comprometido nuestro juicio acerca de los sub-procesos. En el ejemplo de la recolección de basura, se pueden considerar varias opciones. Una es no detallar las actividades relacionadas con la recolección de la basura, mientras otra opción mas extrema sería definir en forma detallada la secuencia de actividades relacionadas con la parada del vehículo en la vía; es decir, decidir si se recolecta una bolsa o un contenedor de basura,

tocar el timbre si es necesario que las personas hagan algo, usar el elevador del camión basurero para cargar la basura o no, y finalmente, continuar con el vehículo hacia otro lado. Aunque este último escenario del ejemplo es exagerado, este es presentado con fines ilustrativos.

La literatura identifica categorías de procesos que permiten discutir este aspecto con más detalle. Estos incluyen los procesos “tightly”, “loosely”, “ad-hoc” (todos ellos del tipo framed) y “unframed”[1]; se dejan estos términos en inglés para no tergiversar su significado.

Tales categorías presentan una variedad de requerimientos diferentes para BPM geo-referenciados. Por ejemplo, con procesos del tipo “tightly”, se requieren de mecanismos que reusen pequeñas piezas de procesos, llamados worklets [7]. Los procesos “loosely” no hacen cumplir el control del flujo, es decir todas las actividades están disponibles para los usuarios [8]. Los procesos “ad-hoc” requieren determinar en cada etapa, lo siguiente por hacer. Y en los procesos “unframed” se requiere algún mecanismo informal de comunicación, como una base mínima para la colaboración. Nosotros sugerimos los siguientes requerimientos para tratar estos aspectos:

R4. Debería ser posible asociar sub-procesos “unframed” a un lugar/región donde los mensajes informales sean intercambiados entre los participantes de estos sub-procesos.

R5. Ciertos tipos de sub-procesos “tightly”, “loosely”, y “ad-hoc”; debieran evolucionar de acuerdo a las tareas y el flujo de control especificado en tiempo real.

R4 y R5 resaltan las diferentes necesidades de mensajes. En ciertos casos mensajes semi-estructurados con eventos de control de flujo y restricciones tiene que ser intercambiados; mientras que en otros, se necesitan mensajes no estructurados. Algunos de estos mensajes incluyen la comunicación entre los usuarios, y otros entre los usuarios y el sistema. Se puede afirmar que los tipos requeridos de comunicación pueden ser soportados a través de una plataforma de microblogging, con su característica principal de envío y recepción de mensajes cortos de y hacia una variedad de destinos mediante un mecanismo de direccionamiento de estos muy simple [9]. En nuestro caso, un requerimiento crítico está relacionado con la geo-referenciación; el que está expresado en el siguiente último requerimiento:

R6. Un sistema ad-hoc BMP geo-referenciado puede ser implementado sobre una plataforma de comunicación del tipo microblogging, el que provee los mensajes a intercambiar

En la sección 4, se discutirá la implementación de la herramienta basada en estos requerimientos.

III. TRABAJOS RELACIONADOS

3.1 BPM Social

La intersección entre BPM y microblogging empezó a recibir atención muy recientemente; llamándose a esta, BPM Social [10,11]. Un ejemplo de ello es Tweetflows [12], una plataforma simple de soporte a la coordinación de procesos de negocios utilizando Twitter. La plataforma tiene el propósito

de juntar tareas y servicios utilizando el método ad-hoc, donde no existe ningún modelo de proceso previo, y el flujo de control es determinado en tiempo real. Los autores identificaron un conjunto de primitivas que soportan el comienzo y fin de la actividad; además de otras, como la de delegar, la cual provee una flexibilidad inusual para empezar un proceso. La plataforma utiliza el típico símbolo “@” para identificar a quienes reciben los mensajes y “hashtags” para identificar servicios. Un problema indicado por los autores es la falta de privacidad y seguridad, puesto que los mensajes son visibles a todos los seguidores. Recientemente, esta plataforma ha sido extendida para dar soporte a workflow móvil [13]; aunque sin dar soporte a actividades geo-referenciadas.

Böhringer [14], también ha considerado el uso de microblogging en BPM, enfocándose otra vez en el proceso ad-hoc, y sugiriendo un estrecho relacionamiento entre este tipo de proceso y varias características de plataformas sociales; tales como el alto grado de libertad que tienen, y el método más proactivo para la selección y ejecución de las actividades. El autor indica que por definición, los procesos no deberían ser modelados, puesto que modelar una sola instancia de trabajo es innecesario y no trae beneficio alguno. El autor presenta el concepto general de un prototipo, utilizando “hashtags” para referir procesos completos y el símbolo “@” para representar las actividades automáticas y la de las personas. El sistema propuesto no incluye el concepto de flujo; y en su lugar los usuarios se coordinan por sí mismos, intercambiando mensajes acerca de un conjunto de actividades. Utilizando “hashtags” es posible recuperar todos los mensajes intercambiados de los procesos.

ACM (Adaptive Case Management) es el acrónimo de una área de investigación relacionada con la adopción de procesos ad-hoc [15]. Un ejemplo de un sistema ACM que integra Social BPM es Casebook [16]. A diferencias de los 2 sistemas mencionados anteriormente, Casebook provee un método más complejo, estructurando actividades ad-hoc alrededor de casos y proveyendo de herramientas específicas para casos de planeamiento, mediciones, aprendizaje, y gestión de catálogos.

3.2 GIS y BPM

Excluyendo casos donde las técnicas de workflow han sido utilizadas para coordinar información geográfica (e.g. [17]), los que están fuera del ámbito de esta investigación, la literatura en cuanto a la integración de BPM y GIS es muy escasa. Kaster et al. [18] y Weske et al. [19] desarrollaron un sistema GIS con un soporte de decisión integrado adoptando un punto de vista de procesos, pero nuevamente quedando fuera del ámbito de esta investigación puesto que no está orientado a procesos de negocios, aunque sí a los del tipo especificados en los requerimientos *R1* a *R3*. Walter [20] sugiere algunas ventajas potenciales al utilizar ambos tipos de sistemas como soporte a la toma de decisiones, por ejemplo en áreas de gestión de incidentes. En la gestión de incidentes, la combinación de procesos “framed” y “nonframed” puede ser beneficiosa, ya que estos requieren regularmente combinar

planeamiento con improvisación. No obstante, no se encontró en la literatura ejemplos concretos que implementen explícitamente los requerimientos *R1* a *R3*.

IV. IMPLEMENTACIÓN

4.1 Control de flujo

Van der Aalst et al. [3] sugirió 20 patrones que cubren la mayoría de las necesidades de control de flujo de BPM. Algunos de estos patrones son muy complejos, y aun no tienen el soporte adecuado por parte de los actuales sistemas BPM, mientras que otros han sido implementados en la mayoría de los sistemas BPM. Considerando la naturaleza exploratoria de esta investigación, optamos por trabajar con un conjunto mínimo de patrones: patrón 1 secuencia (sequence), patrón 2 bifurcación paralela (parallelsplit), patrón 3 sincronización (synchronization), y patrón 11 terminación implícita (implicittermination).

Estos patrones pueden ser implementados de diferentes maneras según el tipo de proceso “framed” considerado (ver sección 2). Así por ejemplo, si el usuario *U1* completa la tarea *T1* y el control de flujo “sequence” tiene que ser seguido con una tarea *T2* hecha por un usuario *U2*; entonces varias posibilidades pueden darse:

1. *U1* notifica a la máquina de workflow que *T1* ha terminado. El worklet por tanto, determina que *T2* debería ser realizado por *U2*.
2. *U1* notifica a la máquina de workflow que *T1* ha terminado; luego la máquina indica que *T2* puede ser ejecutado por *U2* o cualquier otro usuario.
3. *U1* notifica a un agente privilegiado (moderador) que *T1* ha terminado. El agente determina que *T2* debería ser ejecutado por *U2*.
4. *U1* decide que *T2* debería ser ejecutado por *U2* y lo notifica.
5. *U1* decide que *T2* debería ser lo siguiente a ejecutar y notificar a los usuarios de que *T2* está autorizado, para lo cual *U2* puede ofrecerse para ejecutarla.
6. *U1* notifica a los usuarios que *T1* ha terminado. Los usuarios pueden discutir para determinar que *T2* debe ser ejecutado por *U2*, o tal vez *U2* se ofrece para ejecutar *T2*.

Control de flujo	Proceso Ad-hoc framed	Proceso Unframed
“Sequence”	$U \rightarrow U$	$G \rightarrow U$
“Parallel split”	$U \rightarrow U+$	$G \rightarrow U+$
“Synchronization”	$U \rightarrow U+ - U$	$U+ \rightarrow G$

TABLA I.
MECANISMOS DE CONTROL DE FLUJO ADOPTADOS

$U+$ significa “uno o más usuarios”; G significa “grupo”; y \rightarrow significa “transición”. Es decir, $U \rightarrow U$ indica que “se para el control de flujo de un usuario a otro”.

La opción 1 refleja el comportamiento típico de un proceso “framed tightly”. La opción 2 implementa un proceso “framed loosely”. Opciones 3 a 5 reflejan diferentes alternativas para implementar el comportamiento típico de procesos “framed ad-hoc”. Y la opción 6 está asociada con procesos del tipo “unframed”. Analizando las opciones de arriba con más detalle, decidimos no implementar la opción 3, puesto que centraliza el flujo de control en decisiones que requieren mucha carga para un agente privilegiado. También se descarta la opción 5, por ser similar a la estrategia de ejecución de procesos unframed.

Las posibilidades discutidas arriba para el patrón “sequence”, pueden ser extendidas para otros patrones sin mayores problemas; excepto el patrón de “synchronization” en el contexto de un proceso “framed ad-hoc”. La cuestión es quien sincroniza las tareas en un contexto ad-hoc. Considerando el ejemplo donde T1 es hecho por U1, y T2 hecho por U2, debería sincronizarse después el comienzo de T3 hecho por U3. Lo que lleva a varias posibilidades a considerar para el control de flujo en un proceso framed ad-hoc:

1. U1 y U2 notifican a un agente privilegiado de que sus respectivas tareas han terminado. El agente determina que T3 debería ser ejecutado por T3.
2. U1 decide que T2 debería ser mezclado y notifica a U2. Después de recibir una notificación de U2, U1 decide que T3 debería ser ejecutado por U2 y lo notifica. Simétricamente la misma decisión puede ser realizada por U2.
3. La última alternativa es más compleja, ya que requeriría considerar el patrón “parallelsplit” que originó el flujo paralelo para ser sincronizado. Aquí se puede considerar de que el sistema requerirá que el usuario que específico un patrón “parallelsplit”, especifique también la sincronización correspondiente.

Analizando estas 3 alternativas, notamos que teniendo un agente privilegiado que realice las decisiones del control de flujo está en contra del espíritu del método ad-hoc que se quiere implementar acá. Esto último, aplica también para la última alternativa, ya que requiere que los usuarios predeterminen las sincronizaciones cuando definen “parallelsplits”. Estas opciones, violan el principio de procesos “ad-hoc framed”. Por lo tanto, se decide implementar la opción 2 descrita arriba. La Tabla 1 resumen los mecanismos de control de flujo adoptados.

4.2 Mensajes para procesos “ad-hoc” y “unframed”

Según lo discutido previamente, la comunicación informal y el control de flujo requieren de mensajes informales y semi-formales respectivamente. Nosotros adoptaremos un mecanismo tipo Twitter para implementar los mensajes, utilizando el típico “#hashtag” para referirse a instancias de procesos de negocios, y el símbolo “@user” para referirse a los usuarios, incluyendo a la máquina de workflow. El tag “%numero” será adoptado para referirse a los patrones de workflow de van der Aalst, de modo que %1 se refiere a “sequence”, %2 a “parallelsplit”, etc. La especificación de

la sintaxis de los mensajes está descrita en la Tabla 2 mediante el uso de expresiones regulares. Notar que las localizaciones georeferenciadas serán especificadas en la interfaz de la aplicación, y por tanto no son parte de la sintaxis de los mensajes.

TABLA II.
MENSAJES TIPO TWITTER NECESARIOS PARA IMPLEMENTAR LA HERRAMIENTA AD-HOC BPM GEO-REFERENCIADA

Mensajestwitter = código? comentario código= sequence and-split and-join-start and-join-end termination sequence = %1 proceso agente? and-split = %2 proceso flujo agente and-join-start = %3a proceso flujo and-join-end = %3b proceso flujo agente termination = %11 proceso flujo? proceso = #w flujo = \^wagente = @w comentario = .*
--

Una característica interesante del esquema de mensajes adoptado es que no nos referimos explícitamente a las tareas pero si a los usuarios. Una tarea es implícitamente definida mediante el envío de un mensaje a un usuario. Otra característica es que los flujos “parallel” iniciados por un patrón “split” son explícitamente llamados con el tag “^flow”. En la Tabla 4 se ejemplifica como trabaja el esquema de mensajes con el ejemplo de un mesón de ayuda que recibe un mensaje de un usuario ‘para que le ayuden con el mal funcionamiento de su computador.

TABLA III
MENSAJES PARA ADMINISTRAR UN PROCESO AD-HOC DE PETICIÓN DE AYUDA POR FALLA DE COMPUTADOR A UN MESÓN DE AYUDA AL CLIENTE

Order	Sent from	Message
1	client	%1 #HelpDesk my computer has failed
2	client	%1 #HelpDesk @helpdesk need help
3	helpdesk	%1 #HelpDesk we are looking to the problem
4	helpdesk	%2 #HelpDesk ^check @net_team ^ticket @trouble_ticket please check remote access
5	net team	%1 #HelpDesk ping works
6	trouble ticket	%1 #HelpDesk ticket open, 002355
7	net team	%1 #HelpDesk proxy failed
8	net team	%1 #HelpDesk rebooting proxy
9	net team	%1 #HelpDesk @helpdesk done
10	helpdesk	%3 #HelpDesk ^ticket problem solved
11	trouble_ticket	%3 #HelpDesk @helpdesk ^ticket trouble ticket closed
12	helpdesk	%11 #HelpDesk process closed

El proceso #HelpDesk se inicia por un cliente que manda un mensaje diciendo que su computador ha fallado. El cliente entonces manda un mensaje a @helpdesk diciendo que necesita ayuda. El control de flujo es pasado entonces @helpdesk. Note que la tarea es pasada implícitamente a @helpdesk, aunque no esté nombrado.

El usuario @helpdesk manda un mensaje reconociendo el problema. El cliente no se nombra en el mensaje pues esto hubiese significado traspasar el control del flujo al cliente. El cliente es notificado por medio del seguimiento de los mensajes relacionados

al proceso #HelpDeskprocess, como es usual en Twitter.

Entonces @helpdesk bifurca el proceso en dos flujos paralelos, uno que concierne al chequeo de la red, el cual se le da el nombre de ^check, y otro que usa un software de tiquetes de problemas para registrar la queja la que se nombra ^ticket. El flujo llamado ^checkes asignado a @net_team, quien subsecuentemente va a reportar los resultados de varias pruebas de redes, los que incluyen revisar si es posible mandar un mensaje (ping) a la máquina del cliente y probar si el computador proxy está funcionando. En el mensaje 7, el usuario @net_team reporta que el proxy ha fallado. En el mensaje 8, @net_team reporta que el problema ha sido resuelto y traspasa el flujo a @helpdesk.

Entonces tenemos una situación en la cual para flujos paralelos deben ser sincronizados. Esto debe hacerse en dos etapas: En la primera (mensaje 10) @helpdesk manda un mensaje de sincronización (patrón número 3, and-join) al actual dueño del flujo ^ticket flow. Nótese que un mensaje de sincronización comprende por lo menos dos flujos: Si solo uno es especificado, se subentiende que el otro corresponde al flujo actual de quien manda el mensaje. En la segunda etapa, en el mensaje 11, el usuario @trouble_ticket envía un mensaje de sincronización (patrón 3, and-join) al usuario @helpdesk, lo que significa que la sincronización está completa. Finalmente en el mensaje 12 el usuario @helpdesk termina el proceso.

V. EVALUACIÓN PRELIMINAR Y DISCUSIÓN

Una técnica de evaluación basada en escenarios fue utilizada para evaluar la viabilidad del ejemplo y prototipo de aplicación descrito en la sección 4. La ejecución con varios usuarios permitió entender la usabilidad y utilidad percibida, los problemas de coordinación de comunicación entre los usuarios, y los cambios de diseño a introducir. Se tomaron en cuenta 4 evaluadores, quienes seguían los mensajes de la Tabla 3 en forma detallada. Luego, se condujo una serie de entrevistas post sesión, enfocándose en el mecanismo de coordinación.

La evaluación formativa permitió entender varios aspectos de diseño: a) es mejor utilizar un conjunto minimalista de patrones de workflow, para mantener el sistema simple, y entendible para los que no están familiarizados con BPM; b) la asociación de mensajes a las localizaciones pueden no ser requeridas cuando las tareas tienen lugar en la misma localización, pero si cuando estas son distribuidas; c) existen varias maneras para especificar la coordinación de los procesos de los bomberos. Algunas utilizan mensajes informales, tipo conversación. En otras los usuarios deben acordar algún mecanismo explícito de coordinación. d) El mecanismo de mensajes, permite a los usuarios un nivel de detalle suficiente para cumplir las tareas; aunque algunos usuarios sugieren un mayor nivel de detalle

REFERENCIAS

- [1] van der Aalst W (2013) Business Process Management: A Comprehensive Survey. ISRN Software Engineering
- [2] Weber B, Reichert M, Rinderle S (2008) Change patterns and change support features – Enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering* 66 (3):438-466
- [3] van der Aalst W, Hofstede A, Kiepuszewski B (2003) Workflow Patterns. *Distributed and Parallel Databases* 14:5-51
- [4] Goodchild M (2010) Twenty years of progress: GIScience in 2010. *Journal of Spatial Information Science* 1:3-20

- [5] Antunes P, Sapateiro C, Zurita G, Baloian N (2010) Integrating Spatial Data and Decision Models in a E-Planning Tool. In: *Groupware: Design, Implementation, and Use. 16th CRIWG Conference on Collaboration and Technology, Maastricht, The Netherlands, vol 6257.* Springer, Heidelberg, pp 97-112
- [6] Antunes P, Zurita G, Baloian N, Sapateiro C (2014) Integrating Decision-Making Support in Geocollaboration Tools. *Group Decision and Negotiation* 23 (2):211-233
- [7] Adams M, Hofstede A, Edmond D, Van der Aalst W (2006) Worklets: a service-oriented implementation of dynamic flexibility in workflows. In: Meersman R, Zahir T (eds) *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006, vol 4275.* Springer, Heidelberg, pp 291-308
- [8] van der Aalst W, Weske M, Grunbauer D (2005) Case handling: A new paradigm for business process support. *Data & Knowledge Engineering* 53 (2):129-162
- [9] Honey C, Herring S Beyond microblogging: Conversation and collaboration via Twitter. In: *42nd Hawaii International Conference on System Sciences, Hawaii, 2009.* IEEE, pp 1-10
- [10] Brambilla M, Fraternali P, Vaca C (2012) BPMN and design patterns for engineering social BPM solutions. In: *Business Process Management Workshops.* Springer, Heidelberg, pp 219-230
- [11] Erol S, Granitzer M, Happ S, Jantunen S, Jennings B, Johannesson P, Schmidt R (2010) Combining BPM and social software: contradiction or chance? *Journal of software maintenance and evolution: research and practice* 22 (6-7):449-476
- [12] Treiber M, Schall D, Dustdar S, Scherling C Tweetflows: flexible workflows with twitter. In: *Proceedings of the 3rd International Workshop on Principles of Engineering Service-Oriented Systems, 2011.* ACM, pp 1-7
- [13] Treiber M, Schall D, Dustdar S, Scherling C Creating mobile ad hoc workflows with Twitter. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing, 2012.* ACM, pp 1998-2000
- [14] Böhringer M (2011) Emergent case management for ad-hoc processes: a solution based on microblogging and activity streams. In: *Business Process Management Workshops.* Springer, Heidelberg, pp 384-395
- [15] Motahari-Nezhad H, Swenson K Adaptive Case Management: Overview and Research Challenges. In: *IEEE 15th Conference on Business Informatics (CBI), 2013.* IEEE, pp 264-269
- [16] Motahari-Nezhad H, Spence S, Bartolini C, Graupner S, Bess C, Hickey M, Rahmouni M (2013) Casebook: A Cloud-Based System of Engagement for Case Management. *IEEE Internet Computing* 17 (5)
- [17] Chen Q, Wang L, Shang Z MRGIS: A MapReduce-Enabled high performance workflow system for GIS. In: *IEEE Fourth International Conference on eScience, 2008.* IEEE, pp 646-651
- [18] Kaster D, Medeiros C, Rocha H (2005) Supporting modeling and problem solving from precedent experiences: the role of workflows and case-based reasoning. *Environmental Modelling & Software* 20 (6):689-704
- [19] Weske M, Vossen G, Medeiros C, Pires F Workflow management in geoprocessing applications. In: *Proceedings of the 6th ACM international symposium on Advances in geographic information systems, 1998.* ACM, pp 88-93
- [20] Walter M Situational Awareness for Enhanced Incident Management (SAFE-IM). In: *Military Communications Conference, 2007.* IEEE, pp 1-6.



Pedro Antunes es profesor asociado en la Universidad de Wellington, en Nueva Zelanda. Sus intereses abarcan el diseño, desarrollo y valoración de tecnologías colaborativas, enfocándose en particular en interacciones socio-técnicas complejas tales como reuniones electrónicas y toma de decisiones y negociación. Ha publicado más de 70 artículos de este tema y ha participado en más de 40 comités de programa de conferencias. Es Miembro del Comité de Gobierno de CRIWG – International Workshop on Groupware. Desde el año 2000, él también ha trabajado como evaluador de proyectos para la Comisión Europea en múltiples temas claves: eHealth, ambientes de trabajo colaborativos, nuevos métodos de trabajo, y educación y aprendizaje.



Nelson Baloian recibió en el año 1987 en la Universidad de Chile, el título de Ingeniero en Computación. Luego, en el año 1997, obtuvo su PhD. en la Universidad de Duisburg, en Alemania. Su principal foco de investigación corresponde a Sistemas de Aprendizaje Colaborativos Computacionales, tema del cual ha sido autor de más de 20 artículos y gracias al cual ha participado en más de 80 conferencias. Ha sido

profesor visitante en la Universidad de Waseda, en Japón y en la Universidad de Duisburg-Essen, en Alemania. Actualmente, es profesor asociado en el departamento de Ciencias de la Computación, en la Universidad de Chile en Santiago, Chile.



Gustavo Zurita tiene un doctorado en Ingeniería Civil, con mención en Ciencias de la Computación, de la Universidad Católica de Chile. Actualmente es profesor asociado en el Departamento de Control de Gestión y Sistemas de Información de la Facultad de Economía y Negocios de la Universidad de Chile. Su campo de interés abarca temas como el aprendizaje colaborativo móvil, construcción de

conocimiento colaborativo y geo-colaboración. Ha sido autor de más de 50 artículos de conferencias y más de 20 artículos publicados en revistas indexadas a ISI. Algunos de sus artículos del tema de apoyo al aprendizaje colaborativo móvil están entre los más citados en el mundo.