

Sincronización y distribución de tareas para workflows distribuidos en sistemas móviles de información

Task Distribution and Synchronization for Distributed Workflows in Mobile Information Systems

CRISTHIAN FIGUEROA

Estudiante del programa de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca. Integrante del grupo de investigación.

Correo electrónico: cfigmart@unicauca.edu.co

ANDRÉS GUERRERO

Estudiante del programa de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca. Integrante del grupo de investigación.

Correo electrónico: anguerrero@unicauca.edu.co

JUAN CARLOS CORRALES

Doctor (c) Ingeniero en Electrónica y Telecomunicaciones. Universidad del Cauca y Université de Versailles Saint-Quentin-En-Yvelines. Laboratoire D'Informatique Prism.

Correo electrónico: jcorral@unicauca.edu.co

ARMANDO ORDÓÑEZ

Magíster (c) Ingeniero en Electrónica y Telecomunicaciones. Universidad del Cauca. Docente Universidad del Cauca.

Correo electrónico: jaordonez@unicauca.edu.co

MAURICIO MACA

Magíster en Ciencias Matemáticas. Universidad del Cauca y Universidad del Valle. Docente Universidad del Cauca.

Correo electrónico: mmaca@unicauca.edu.co

Clasificación del artículo: reflexión

Fecha de recepción: 13 de abril de 2007

Fecha de aceptación: 19 de julio de 2007

Palabras clave: sistemas móviles de información, procesos de negocios, algoritmo de hormigas, grafos, workflow, web service, peer to peer, BPEL.

Key words: mobil systems of Information, business processes, ants algorithm, graphs, workflow, peer to peer processes, BPEL.

RESUMEN

En este artículo se presenta la arquitectura de sincronización de BDMobIS, un sistema para la distribución de los procesos de negocios entre dispositivos

móviles, a fin de reducir los tiempos de ejecución y las comunicaciones entre los participantes. La sincronización es uno de los puntos más críticos, puesto que es la responsable de que el flujo normal

del proceso no se pierda aún después del particionamiento. Se propone un mecanismo de sincronización y distribución basado en la inserción de actividades de sincronización que permiten mantener el flujo de ejecución, compartir e intercambiar la información global del proceso basada en Servicios de Mensajería Corta. Además, en este artículo se presenta un caso de estudio para la ejecución del proceso de matrícula de la Universidad del Cauca.

ABSTRACT

This paper shows the synchronization architecture of BDMobIS, a system for the distribution of busi-

ness processes between mobile devices in order to reduce execution times, and the amount of communications among its participants. Synchronization is one of the most critical stages because it is responsible for keeping the normal flow of the process, even after its partitioning stage. A mechanism of synchronization and distribution is proposed, based on the insertion of synchronization activities which permit to keep the flow of execution, to share and exchange global information of the process, based on Short Messages Services. This article also presents a Study Case for the process of tuition fee payment at the university called "Universidad del Cauca".

* * *

1. Introducción

Las organizaciones necesitan de una gestión eficiente de los procesos internos y la consecuente optimización de los recursos. Para conseguir este objetivo es vital una coordinación entre los trabajadores y que el tiempo que se destine para cada actividad sea el mínimo necesario a fin de evitar retardos en la ejecución de los procesos de la empresa. La propuesta aquí descrita se enfoca en mejorar estos aspectos, haciendo uso de los sistemas móviles de información (MobIS), para realizar parte de las tareas del Workflow de la empresa, y así optimizar los tiempos de ejecución de los procesos. Hoy en día, los empleados necesitan estar en constante movilidad, de manera que cada trabajador debería poder gestionar las tareas de los procesos empresariales que le corresponden, desde su dispositivo móvil, ahorrando de esta manera tiempo y contando con información de sus tareas en cualquier momento y en cualquier lugar. Visto de este modo los dispositivos móviles de los empleados se convierten en sistemas móviles de información.

Un proceso de negocio, según [1], se define como la interacción entre participantes y la ejecución de actividades, de acuerdo con un conjunto de reglas definidas, a fin de alcanzar un objetivo en común. La

automatización de estos procesos, es lo que se conoce como Workflow, lo cual según [2], se define como el conjunto de actividades que coordinan personas o software. Estos flujos de trabajo, o Workflows, pueden estar representados por máquinas de estado, redes de Petri, autómatas finitos, grafos, entre otros, para su análisis matemático o visual.

Otras propuestas como [3] presentan un sistema que integra dispositivos móviles a la ejecución Workflows; sin embargo, este tipo de arquitecturas plantean la asignación de responsabilidades individuales, de manera que cada dispositivo móvil tiene a su cargo la ejecución de una única tarea dentro del flujo de trabajo especificado. Estos sistemas cuentan con algoritmos para la asignación de responsabilidades a los diferentes dispositivos móviles, basando su funcionamiento en restricciones tales como que un trabajador no podrá tener asignadas dos tareas, cuyos tiempos de inicio y fin sean los mismos, o que dos trabajadores no pueden tener asignadas dos tareas consecutivas si el segundo trabajador no conoce los resultados obtenidos por el primer trabajador tras ejecutar la primera.

La presente propuesta se basa en la distribución de subflujos de trabajo en los dispositivos, de manera que se requiere una constante comunicación entre los

diferentes trabajadores de la organización. Precisamente, en BDMobIS se pretende minimizar las comunicaciones necesarias para la coordinación de los procesos de negocios distribuidos, por medio de la aplicación de un algoritmo de particionamiento de grafos que busca la disminución de las comunicaciones entre los dispositivos móviles obteniendo como consecuencia la reducción de costos para las organizaciones.

Otros trabajos, como [4] [5], proponen un particionamiento de grafos, los cuales son representados por estructuras UML (Lenguaje Unificado de Modelado), en los cuales la validación se hace gracias a la gramática de grafos atribuidos (sus siglas en inglés AGG) [6]. Nuestra propuesta incluye un mecanismo de optimización de la comunicación que se hace al aplicar algoritmos de particionamiento de grafos que reducen al máximo las interacciones entre los diferentes trabajadores de la organización.

El resto de este artículo se organiza de la siguiente manera: la segunda parte ofrece la motivación hacia nuestra investigación sobre el tema, posteriormente, en la tercera sección, se presenta una descripción general de la arquitectura de BDMobIS. La cuarta sección describe el módulo de particionamiento de procesos, basado en un algoritmo de colonia de hormigas para grafos, la quinta, por su parte, detalla el proceso de sincronización de las tareas distribuidas, la sexta describe el proceso de distribución de los subprocesos en sistemas móviles de información. Finalmente, se plantean las conclusiones y los trabajos futuros.

2. Motivación

Hoy en día las organizaciones requieren de la intervención de muchos participantes en sus procesos de negocios, algunos de los cuales involucran una gran movilidad. Esta situación origina enormes retardos, porque los procesos requieren que los participantes permanezcan en línea con el motor de ejecución, para realizar y sincronizar las tareas que les corresponden. Una alternativa tecnológica es la distribución de los procesos entre los dispositivos

móviles, cada vez más potentes de los empleados. Esta solución requiere de mecanismos robustos de sincronización entre las partes mediante una constante comunicación entre todos los dispositivos y el motor central, lo cual, evidentemente, implica un consumo amplio de ancho de banda, que en muchas ocasiones se hace inviable debido a los costos asociados.

3. Arquitectura de BDMobIS

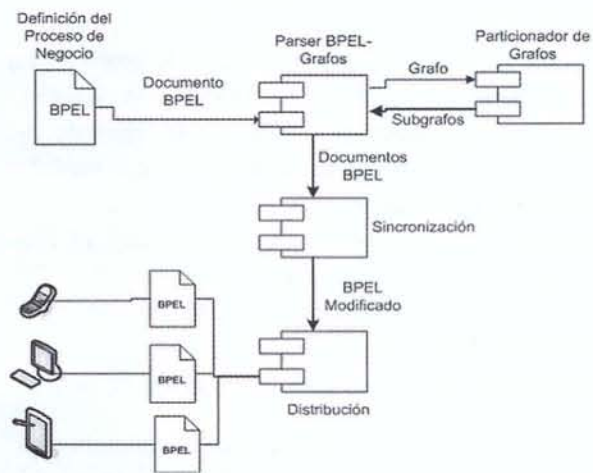


Figura 1. Arquitectura BDMobIS

Con el objetivo de reducir el problema de partición de procesos de negocios a un problema de particionamiento de grafos, el módulo *Parser BPEL-Grafos* (ver figura 1) transforma un documento BPEL a un grafo y viceversa. El módulo *Particionador de grafos* contiene el algoritmo de particionamiento que se aplica al grafo BPEL obtenido por el módulo anterior. Las partes del grafo BPEL entregadas por el algoritmo particionador son transformadas, nuevamente, a un archivo BPEL, a fin de facilitar su ejecución en los motores BPEL instalados en los dispositivos móviles. Finalmente, el módulo de *distribución* distribuye a los dispositivos móviles las subpartes del proceso BPEL obtenidas por el particionamiento. El módulo *sincronizador* se encarga de insertar tareas de sincronización al proceso particionado, para mantener correcto el flujo de

ejecución. Por otro lado, el módulo *distribuidor* realiza el proceso de reconfiguración en caso de que uno de los dispositivos móviles sea cambiado o esté apagado, lo que se detallará posteriormente.

4. Analizador de documentos para cambio de notación de BPEL a grafos

La primera función de BDMobIS es el particionamiento de los procesos de negocio que se compone de los módulos de conversión de procesos BPEL a grafos, el particionamiento de grafos y el módulo de transformación de grafos a XML o BPEL. El módulo de conversión de procesos BPEL a grafos, recibe la definición de un proceso de negocio. La implementación actual de BDMobIS emplea BPEL, pero no está restringido sólo a este lenguaje. Se seleccionó BPEL por su flexibilidad para modelar un proceso de negocio, en comparación con otros lenguajes como BPML (Lenguaje para el Modelamiento de Procesos de Negocio) y WSCDL (Lenguaje para la Descripción de la Coreografía de Servicios Web). En [7] se resumen los criterios que se tuvieron en cuenta para la selección de BPEL.

La definición del proceso se hace atendiendo rigurosamente los requerimientos de una determinada organización, la cual manifiesta querer automatizar un proceso que repetidamente se ejecuta. Al documento BPEL se le aplica el algoritmo de particionamiento multiagente y, una vez realizado este proceso, se crean nuevos documentos XML para enviar a los dispositivos móviles, basados en la estructura gráfica de fragmentación arrojada por el algoritmo bioinspirado. Estas definiciones corresponden al número de módulos que hayan sido especificados en la división del proceso de negocio. Por ejemplo, si hay un cliente y dos socios del proceso, el número de documentos BPEL que deben ser generados para esta parte del proceso son cuatro, correspondientes al cliente, los dos socios del proceso, más un módulo adicional, que aliviana y facilita la carga de nodos en los demás módulos gracias a que éste se ejecutará en un computador de escritorio con buenas características de procesamiento. Todas estas definiciones se hacen con la

ayuda de un entorno integrado de desarrollo que genera documentos BPEL los cuales son ejecutados tanto por dispositivos móviles como por motores de ejecución BPEL de escritorio. Cada módulo generado en esta parte del sistema tiene que ser cuidadosamente elaborado, para que el flujo de ejecución no se pierda y el objetivo de negocio se pueda conseguir.

Una vez elaborado el modelo del cual se crearán las múltiples instancias del proceso, se procede a transformar su notación, a fin de realizar su particionamiento. Dicha transformación se lleva a cabo si se utiliza la propuesta especificada en [8], que toma como entrada un archivo BPEL y genera un modelo BPEL representado por objetos Java.

Para la arquitectura propuesta es necesario utilizar un algoritmo que realice la división de un “workflow” centralizado en varios “subworkflows”, distribuidos entre los dispositivos asociados al sistema. Para esto el “workflow” se representa a través del lenguaje de los grafos, dado que para muchas aplicaciones en el cálculo científico, el problema de descomponer un cómputo entre varios procesadores se puede describir convenientemente de esta forma [10]. Una vez obtenida la representación del “workflow” como grafo, se procede a realizar un proceso conocido como particionamiento de grafos, que consiste en dividir un grafo en subgrafos o subconjuntos de vértices y aristas, de tal manera que las aristas entre los subgrafos (que representarán las comunicaciones entre los dispositivos móviles) conocidas como aristas de corte sean las mínimas posibles, para así reducir el número de comunicaciones entre los procesos involucrados en cada subgrafo.

Existen muchas formas de realizar particionamiento de grafos, un proceso muy conocido por su gran complejidad, ya que está ubicado en la categoría de problemas tipo NP completo [11]. Para este módulo se empleó el algoritmo bio-inspirado en colonias de hormigas [13]. Inicialmente, estos algoritmos fueron propuestos por Marco Dorigo [14] [15] [16] y se basan en el comportamiento de hormigas reales en su proceso natural para conseguir alimento.

Una hormiga puede encontrar el camino entre su hormiguero y la fuente de comida sin necesidad de indicaciones visuales. Así pues a cada nodo se le asigna un color que represente el subdominio o subgrafo al cual pertenece. Luego se distribuye un cierto número de hormigas a través de los nodos de un grafo coloreándolos (moviéndolos de un subgrafo a otro) mediante un criterio de optimización local [17]. Una descripción detallada del algoritmo utilizado en el presente proyecto se puede observar en [18].

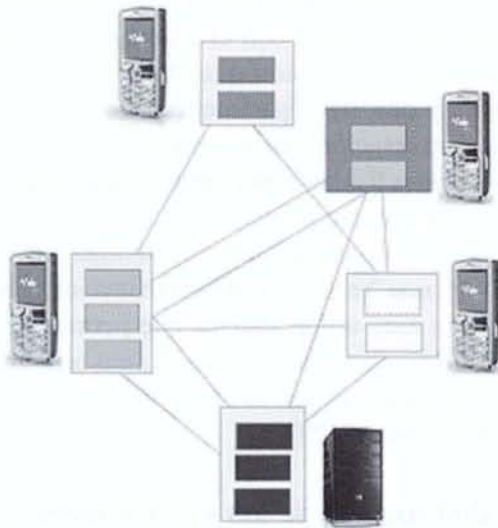


Figura 2. Proceso de particionamiento inicial. Obsérvese que hay 10 aristas de corte

Por ejemplo, en la figura 2 se puede observar un proceso de negocio representado a través de grafos, dicho proceso ya ha pasado por una etapa denominada particionamiento inicial, que toma como criterio de particionamiento los “partner links” del proceso BPEL, es decir, de acuerdo con el número de “partner links” se crea un determinado número de módulos que es igual al número de “partner links” más uno que corresponde al servidor central, el cual, al igual que los otros módulos, ejecutará un subproceso BPEL, para el caso concreto de la figura 2 se pueden observar cinco módulos con diez aristas de corte, o aristas que los unen entre sí.

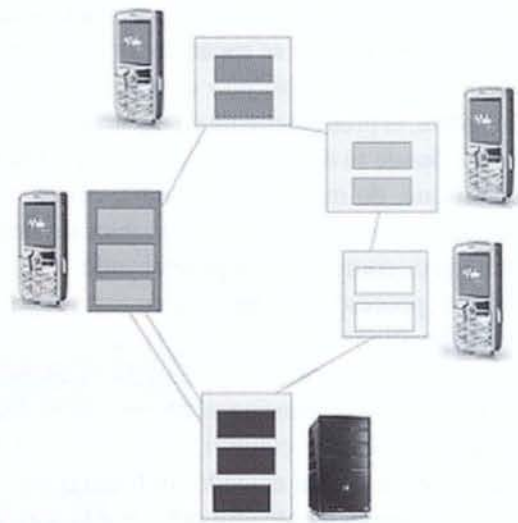


Figura 3. Grafo después del proceso de particionamiento realizado con el algoritmo de colonia de hormigas. Sólo queda con seis aristas de corte de las diez iniciales vistas en la figura 2

Después de aplicar el algoritmo de particionamiento descrito en [18], el cual toma como punto de partida el proceso de particionamiento inicial, se obtienen la figura 3 que presenta el mismo número de módulos que la figura 2, pero esta vez con el número de aristas de corte reducidas de las diez iniciales a tan solo seis aristas de corte. Esto se logra gracias a la acción repetitiva del algoritmo de particionamiento ya nombrado.

El algoritmo de particionamiento permite la reducción de aristas de corte, que representan las conexiones entre los módulos, las cuales implican comunicaciones entre los dispositivos móviles. De esta forma, la reducción en el número de aristas representa una disminución en costos relativos a comunicación.

5. Sincronización de los subprocesos en sistemas móviles de información

La sincronización en BDMobIS es uno de los puntos más críticos del sistema. Ella es la responsable de que el flujo normal del proceso no se pierda aun después del particionamiento. Cuando el proceso de

negocio se define por primera vez, las actividades básicas y estructuradas se encargan de mantener su flujo, pero, una vez aplicado el particionamiento multiagente, el proceso pierde su estructura normal y es aquí donde se debe sugerir una idea de sincronización que, de manera consistente, conserve el estado del proceso junto con sus variables siguiendo un método de comunicación generalizado. Según lo anterior, el proceso se divide y se envían sus partes a los dispositivos móviles para su ejecución. Si entre ellos no existe una comunicación eficiente, el sistema pierde su secuencia y el objetivo de negocio se verá imposible de cumplir, por esta razón los participantes se deben entender utilizando medios eficientes y ampliamente utilizados por la mayoría de los dispositivos móviles.

Para el proceso de sincronización se utilizó el mecanismo descrito en [19], en el que se argumenta que la inclusión de actividades de sincronización permite mantener la consistencia del proceso de negocio distribuido.

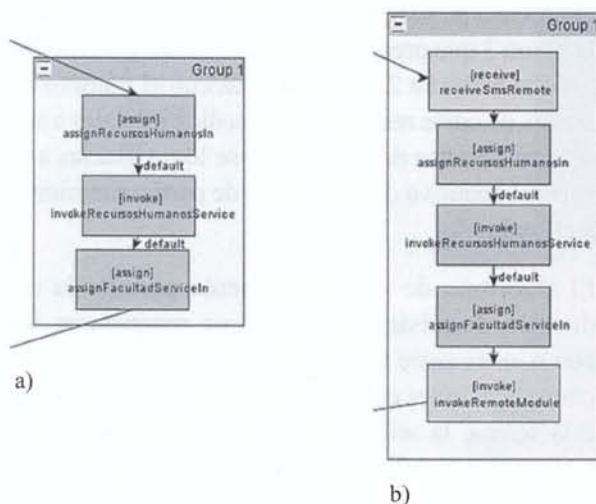


Figura 4. Tareas de sincronización. En el literal a) se puede apreciar un módulo de particionamiento sin actividades de sincronización. En el literal b) se puede apreciar la inserción de tareas de sincronización, siendo éstas la primera (receiveSmsRemote) y la última del módulo (invokeRemoteModule)

En la figura 4 se puede apreciar la inclusión de tareas de sincronización. La gráfica de la izquierda posee tres nodos correspondientes al proceso de negocio original al que se le ha aplicado el algoritmo multiagente de particionamiento. La gráfica de la derecha posee cinco nodos de los cuales tres de ellos corresponden al proceso original y los dos restantes corresponden a actividades de sincronización insertadas al proceso. La primera actividad de sincronización es una actividad “receive” (receiveSmsRemote), que está encargada de recibir los resultados arrojados por los socios precedentes a este módulo de particionamiento. La segunda es una actividad invoke (invokeRemoteModule); ésta es insertada con el objeto de notificarle al proceso en ejecución que otro actor debe ser llamado y que algún parámetro debe ser incluido en la conversación que se mantenga con él. Las invocaciones llaman a otras aplicaciones, llamadas socios del proceso de negocio, las cuales aportan una funcionalidad única. Una actividad “invoke” junto con una “receive” conforman una interacción asíncrona muy útil al sistema que permite que el proceso de negocio puede ejecutar otras actividades mientras socios asíncronos resuelven lo que se le ha encomendado [20].

6. Distribución de los subprocessos en sistemas móviles de información

Con BDMobIS cada empleado ejecuta desde su terminal un subprocesso de negocio cuyos resultados se pasan de un trabajador a otro, hasta llegar a la consecución satisfactoria del proceso global. A partir de los subprocessos generados en la etapa de sincronización que deben ser ejecutados de manera descentralizada por sus respectivos responsables. Para cada socio del proceso de negocio se genera un documento con las tareas que debe ejecutar dicho participante, los cuales son producto del particionamiento multiagente y del proceso de sincronización. Dichos documentos son almacenados en los dispositivos móviles para reducir las comunicaciones necesarias durante la ejecución del proceso. De esta manera los dispositivos móviles de los empleados

poseen toda la información necesaria para ejecutar las tareas sin depender de los servicios remotos.

Para esta distribución de tareas existen dos alternativas: en [21] se presenta una implementación del XMLPull API llamada kXML2. Este conjunto de clases hacen posible la lectura de documentos XML en dispositivos móviles que cuentan con capacidades de procesamiento reducidas. Esta opción es útil si el texto que se envía a los dispositivos móviles está en el formato XML. Junto con esta opción de lectura, están otros aspectos relacionados tales como la invocación de los servicios Web necesarios, la comunicación del proceso central con los diferentes dispositivos móviles y la conservación de las variables globales del proceso de negocio a lo largo de toda su ejecución en los sistemas móviles de información.

La invocación de los servicios Web, por parte de los usuarios, se hará detectando la actividad básica “invoke” en el texto XML que esté procesando cada dispositivo móvil. Una vez se identifique la necesidad de un servicio Web, por parte del proceso, el dispositivo mediante protocolo SOAP (sigla en inglés para Protocolo de Acceso Simple a Objetos) hará su invocación si se siguen los lineamientos especificados en su archivo WSDL (sigla en inglés para Lenguaje para la Descripción de Servicios Web).

La segunda propuesta es SLIVER descrita en [22], que propone un motor muy pequeño de orquestación capaz de ejecutar documentos BPEL en dispositivos móviles. Dicho motor es un sistema que ha sido diseñado para ejecutarse, al igual que kXML2, tanto en dispositivos de reducidas capacidades como en computadoras de escritorio. Por este motivo es una herramienta más general que la anterior en cuanto a la gama de dispositivos en los cuales ésta pueda ser instalada.

En la arquitectura propuesta por SLIVER, la capa de transporte es capaz de intercambiar mensajes de objetos en forma de cadenas XML serializadas. Esas cadenas se convierten hacia y desde objetos Java por medio de capas analizadoras de XML y SOAP.

La capa SOAP server envuelve servicios Java, provistos por el usuario. Cuando llegan mensajes desde las capas XML y SOAP, el servidor SOAP los direcciona al servicio solicitado por el proceso de negocio. La respuesta generada se serializa y se envía a través de la red, gracias a las capas XML y SOAP.

Se seleccionó la implementación del API XMLPull kXML2, debido a la simplicidad de su implementación en la Edición Micro de Java. Su ejecución es ágil en los dispositivos móviles de reducidas capacidades de procesamiento, ya que el análisis que dicho API realiza sobre los documentos es manejado por el usuario que, a su vez, controla a aquella que se le debe prestar atención en el documento XML. Otras opciones similares, como las de tipo Push, hacen una lectura de todo el documento sin tener en cuenta si lo que se está procesando es de interés o no para el cliente. Además, según [21], la opción seleccionada no se preocupa por la validación de los documentos cargados por el sistema. Esta característica es muy importante para BDMobIS, ya que los archivos que se le pasan al dispositivo móvil para su análisis son secuencias BPEL las cuales pueden variar en comparación con el lenguaje XML.

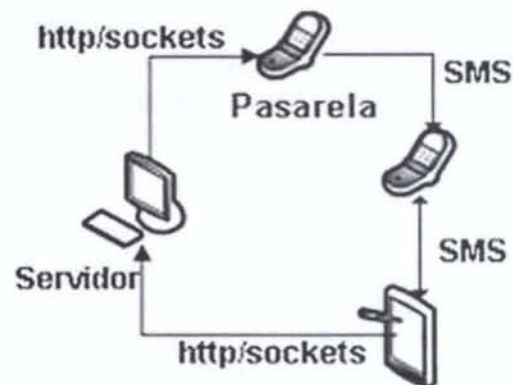


Figura 5. Grafico de distribución

En BDMobIS se han identificado tres tipos de comunicación (figura 5). La primera es la que se ocupa de comunicar un dispositivo móvil con un sistema de

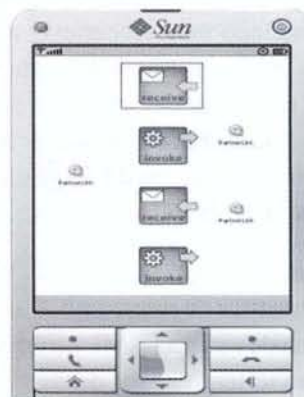
escritorio. Esta comunicación se implementa sobre el protocolo HTTP, gracias a las facilidades de navegación que tengan los dispositivos móviles de los trabajadores de la organización. Dicha conexión se lleva a cabo gracias a la generación de “stubs”, los cuales presentan a los dispositivos móviles toda la funcionalidad de las aplicaciones para invocar. En el caso del presente artículo, Servicios Web, el segundo tipo de comunicación es la que tiene lugar entre dos dispositivos móviles. Para esta comunicación BDMobIS utiliza el servicio de mensajería corta SMS, por ser el protocolo más conocido, aceptado, y compatible con la mayoría de equipos móviles existentes hoy en día. El tercer y último tipo de comunicación identificado es el existente entre una aplicación de escritorio (Servicio Web) y un dispositivo móvil. Para ello BDMobIS utiliza también el envío de SMS. Con fines de desarrollo del prototipo, BDMobIS utiliza una pasarela implementada con un Servicio Web alojado en el servidor y un dispositivo móvil (figura 5). Entre el proceso BPEL y el dispositivo móvil pasarela se utiliza como protocolo de comunicación la implementación de Sockets y entre el dispositivo móvil pasarela y el dispositivo móvil destino se utiliza el envío simple de SMS. En dicha pasarela se optó el uso de “sockets”, gracias a que éstos establecen comunicaciones no orientadas a la conexión lo que repercute en el bajo consumo de medios de conexión. Se sabe que en la práctica se suelen utilizar las estaciones Centro de Servicio de Mensajería Corta (SMSC) y como trabajo futuro, para este tipo de comunicación, se probará a BDMobIS bajo este entorno.

Los diferentes tipos de comunicación arriba nombrados se enfocan en el envío de mensajes cortos de texto. Éstos transportan el valor actual de variables globales al proceso de negocio y provocan la lectura de los módulos BPEL residentes en los dispositivos móviles. Mediante una interfaz gráfica amigable el dispositivo móvil mostrará al usuario sus actividades pendientes en el proceso de negocio y una vez todas éstas estén concluidas, el sistema procede a comunicar los resultados producidos por el trabajador al resto de los participantes.

7. Prototipo desarrollado

Se desarrolló una solución basada en BDMobIS para el caso de estudio de la gestión de los procesos de negocio de certificación de descuento de matrículas para la Universidad del Cauca, un proceso que se realiza más de 300 veces por semestre académico. Dicho proceso consiste en la autorización de un descuento en la matrícula financiera para los estudiantes en determinados casos, como por ejemplo ser hijo de algún funcionario de la institución.

En el proceso intervienen varios participantes que resuelven si el estudiante tiene o no derecho al descuento en su matrícula. Dichos participantes, que figuran como “partnerlinks” en el proceso: son la División de Personal de la Universidad, Caja de Recaudos de la División Financiera y, por último, la Secretaría Académica de cada Facultad. Las variables globales identificadas en este proceso son la cédula y el código, tanto del interesado en el descuento como del funcionario. Con esta información manipulada por las partes especializadas se consigue autorizar o no al estudiante para que pague su matrícula financiera con descuento. Actualmente, el proceso de negocio inicia cuando el estudiante cancela en la División Financiera de la universidad un certificado laboral de su padre funcionario de la institución. Una vez obtenido este certificado, emitido por la División de Personal de la universidad, el estudiante lo lleva a su facultad y, posteriormente, es enviado nuevamente a la división financiera para que sea tramitado su recibo de pago.



a)



b)

Figura 6. Interfaces gráficas de usuario. El literal a) es un ejemplo de cómo se le muestra al trabajador sus deberes dentro del proceso de negocio. El literal b) es un ejemplo de la ejecución de una tarea “workflow” en el dispositivo móvil del trabajador

Siendo así, se tienen cinco submódulos que son el resultado de un proceso de particionamiento. Cuatro de ellos son implementados en consolas móviles y procesan documentos XML y hacen envío y recepción de mensajes cortos de texto SMS. El quinto módulo es un proceso BPEL normal que es ejecutado en un computador de escritorio. Los trabajadores móviles, en su dispositivo móvil, están al tanto del subproceso que se les ha asignado y mediante una interfaz gráfica amigable (ver figura 6), cumplen una a una las actividades de las cuales son responsables en el proceso. Una vez completadas sus tareas, los trabajadores envían un mensaje corto de texto al siguiente módulo con los estados de las variables globales del proceso, para que éste siga con la ejecución del mismo. De esta manera, se completa todo el flujo mostrándole al estudiante interesado la aprobación o desaprobación del descuento en su matrícula financiera.

8. Conclusiones

En este artículo se presenta la arquitectura de sincronización y distribución de BDMobIS, un sistema para disminuir los tiempos de ejecución por medio de la distribución de los subprocesos de negocios entre los dispositivos móviles de los usuarios. La sincronización en BDMobIS es uno de los puntos más críticos del sistema. Ella es la responsable de que el flujo normal del proceso no se pierda aun después del particionamiento.

Se destaca un mecanismo de sincronización basado en la inclusión de tareas en cada subproceso, lo que permite coordinar los flujos de trabajo distribuidos, y guardar una secuencia ordenada de todas las actividades que deben ser ejecutadas en el proceso de negocio. De igual forma, se presentaron los tipos de comunicación en un entorno distribuido de ejecución de “workflow” y las alternativas seleccionadas para BDMobIS.

Como trabajo futuro se tienen pensados varios aspectos. El primero de ellos es la inclusión de un motor de ejecución un poco más genérico, como lo es SLIVER [22]. Este sistema es aplicable tanto a aplicaciones de escritorio como para entornos móviles. Sliver sería una muy buena opción para reemplazar el API kXML2 utilizado actualmente en BDMobIS. Otra futura investigación para BDMobIS es la reconfiguración automática del sistema de distribución de tareas a los dispositivos móviles. Para el sistema es de vital importancia que todos los dispositivos de los trabajadores estén disponibles el mayor tiempo posible. Si un dispositivo móvil sale del área de cobertura o simplemente se encuentra apagado, el sistema debe ser capaz de identificar esta de situación y proponer dispositivos candidatos para la ejecución de las tareas pendientes para el proceso. Esto sería una aproximación hacia las redundancias de dispositivos móviles en BDMobIS.

Referencias bibliográficas

- [1] Arkin, A. and Agrawal, A. (August 2001) *Business Process Modeling Language*. Working Draft 0.4. BPMI.org Intalino, Inc. Status: Working Draft. Draft Unpublished.
- [2] Cor, G. 10. (2006) *Oportunidades arquitecturales para Workflow*. Paradigma Software. Microsoft Regional Director. Uruguay, Paraguay y Bolivia. Draft Unpublished.
- [3] Hackmann, G., Sen, R., Mart, H., Gruia-Catalin, R. and Christopher, G. (2006) *MobiWork: Mobile Workflow for MANETs*, Washington University in St. Louis School of Engineering & Applied Science. Department of Computer Science & Engineering. WUCSE-2006-18. In Press. April 14, pp. 5-12.
- [4] Baresi, L., Maurino, A. and Modafferi, S. (2004) *Workflow Partitioning in Mobile Information Systems*. Politecnico di Milano Dipartimento di Elettronica e Informazione. P. zza L. Da Vinci 32-20133 – Milano Italy, Draft Unpublished. pp. 3-12.
- [5] Maurino, A. and Modafferi, S. (2005) *Partitioning rules for orchestrating mobile Information systems*, Politecnico di Milano Dipartimento di Elettronica e Informazione. Piazza L. Da Vinci 32 – 20133 – Milano Italy. 2005. Pers Ubiquit Comput (20005) 9: 291 – 300. DOI 10.1007/s00779-004-0333-4. Received: 30 July 2004/ Accepted: 10 November 2004/Published online: 17 August 2005. Springer-Verlag London Limited.
- [6] Beyer, M. (1992) AGG1.0 Tutorial, Technical University of Berlin, Department of Computer Science. Draft Unpublished.
- [7] Yushi, C., Wah, L. and Limbu D. (2004) *Web Services Composition –An Overview of Standards* Singapore Institute of Manufacturing Technology. Section Four, pp 10. Draft Unpublished.
- [8] Baloche, I. and Draperi, E. (2006) *Projet Annuel ISTY 3 Transformation BPEL en Graphe*, Draft Unpublished. pp. 5-32.
- [9] Himsolt, M. (1996) *GML: Graph modeling language*, Draft Unpublished.
- [10] Hendrickson, B. and Leland, R. (1995) A Multilevel Algorithm for Partitioning Graphs, Proceeding of Supercomputing. In S. Karin, editor, Proc. Supercomputing. ACM Press, pp. 1-13.
- [11] Hernández, G. (2000) *Complejidad y Grafos*, II Seminario de matemática discreta y codificación de la Informática. Universidad Politécnica de Madrid. Draft Unpublished.
- [12] Cortez, A. (2004) *Teoría de la complejidad computacional y teoría de la computabilidad*. Lima, Perú: Universidad Nacional Mayor de San Marcos. Draft Unpublished.
- [13] Comellas, F. and Sapena, E. (2006) *A Multiagent Algorithm for Graph Partitioning*, Lecture Notes in Comput. Sci. vol. 3907, In Press. pp. 279-85.
- [14] Colorni, A., Dorigo, M. and Maniezzo, V. (1991) Positive Feedback as a Search Strategy. In Press, Tech. Rept. 91-16 Politecnico di Milano - Department of Electronics, Milano – Italy.
- [15] Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italy. Elsevier Publishing.
- [16] Dorigo, M. and Stützle, T. (2004) *Ant Colony Optimization*. MIT Press.
- [17] Comellas F., Ozón J., Cortés A., Abril J. and Vaquer M. (1999) *Sistemas multiagente para la asignación de frecuencias en redes celulares*. IX Jornadas de I+D en Telecomunicaciones, UPC, Barcelona. 17-18.
- [18] Figueroa, C., Guerrero, A., Ordoñez, A., Maca, M., Corrales, J. (Junio de 2007) Distribución de procesos de negocios en sistemas móviles de información basada en un algoritmo de colonia de hormigas. Revista *Avances en Sistemas e Informática*, Vol. 4 (1). Universidad Nacional de Colombia sede Medellín. Aceptado Junio 2007.
- [19] Maurino, A. y Modafferi, S. (November 2004) *Partitioning Rules for Orchestrating Mobile Information Systems*. Politecnico di Milano Dipartimento di Elettronica e Informazione. Piazza L. Da Vinci 32 – 20133 – Milano Italy. Pers Ubiquit Comput (20005) 9: 291 – 300. DOI 10.1007/s00779-004-0333-4. Received: 30 July 2004.
- [20] Andrews, T; Curbera, F. *Business Process Execution Language for Web Services 1.1*. Copyright© 2002, 2003 BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems. All rights reserved.
- [21] Froufe, A. y P. Jorge. (2004) *J2ME Java 2 Micro Edition Manual de usuario y tutorial*. Madrid: RA-MA Editorial pp. 381-397.
- [22] Hackmann G., Haitjema, M., Christopher, G., Gruia-Catalin R. (2006) *Sliver: A BPEL Workflow Process Execution Engine for Mobile Devices*. WUCSE – 2006 – 37. Department of Computer Science & Engineering. University in St. Louis School of Engineering and Applied Science, in Press, pp. 5–14.