

mSALSA: Agentes de Software para el Desarrollo de Aplicaciones Móviles

Irma Amaya¹, Pedro C. Santana², Marcela D. Rodríguez^{1,2}, Jesús Favela¹

¹Departamento de Ciencias de la Computación

Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)

²Facultad de Ingeniería, Universidad Autónoma de Baja California

{amaya, psantana, marcerod, favela}@cicese.mx

Resumen

Los ambientes de cómputo ubicuo se caracterizan por la diversidad de dispositivos a través de los cuales los usuarios móviles pueden acceder a información y servicios. Los usuarios de estos ambientes requieren transferir información de su PDA a cualquier otro dispositivo, o viceversa. Para facilitar la implementación de esta característica en los sistemas de cómputo ubicuo, decidimos crear un componente de migración. La base para el desarrollo de este componente es el middleware SALSA, el cual permite crear sistemas de cómputo ubicuo mediante agentes autónomos. Para la implementación de la versión de escritorio de dicho componente, se utilizó la versión de SALSA basada en Java. Para la versión móvil del componente, se extendió la funcionalidad de SALSA, implementándole un subconjunto de clases desarrollado en el lenguaje de programación C# denominándole mSALSA (micro SALSA). En este artículo, presentamos cómo mSALSA facilitó la implementación del componente de migración e ilustramos su funcionalidad.

Abstract

Ubiquitous computing environments are characterized by the diversity of devices through which the mobile users can accede to information and services. The users of these environments require to transfer information of their PDA to any other device, or vice versa. In order to facilitate the implementation of this characteristic in ubiquitous computing systems, we decided to create a migration component. The base for the development of this component is SALSA middleware, which allows to create ubiquitous computing systems by means of independent agents. For the desktop version implementation of this component, we used a SALSA version based on Java. For the mobile version of the component, the SALSA's

functionality was extended implementing a sub set of classes developed in C#, it version was denominated mSALSA (micro SALSA). In this article, we presented how mSALSA facilitate the migration component implementation and illustrate its functionality.

Palabras clave: middleware, componentes de software, agentes de software, SALSA

1. Introducción

El desarrollo de aplicaciones de cómputo ubicuo debe considerar aspectos de los usuarios como: movilidad, comunicación y colaboración, así como la interacción entre los diversos dispositivos de su ambiente. En este último aspecto, varios trabajos de investigación sugieren que los usuarios de un ambiente de cómputo ubicuo requieren transferir información de un dispositivo a otro. Ejemplo de estos proyectos son: el sistema *SharedNotes* [3], el cual permite la transferencia de información de una pda a una pantalla pública a través de la publicación de notas personales. *MB2Go* [4] utiliza el enfoque de transferencia de URLs de dispositivos personales (pda, laptop, pcs) a pantallas públicas, empleando la selección de la pantalla destino a la cual se desea transferir el URL para ser desplegado. *Passage* [5] realiza la migración de información a través de objetos físicos (denominados passengers) los cuales son utilizados para transportar físicamente la información de una PC a una pantalla pública.

Una característica que presentan recurrentemente los sistemas ubicomp es que necesitan dar soporte a la migración de información entre diversos dispositivos (pda, pantallas, públicas, laptops, entre otros). Basándonos en esta característica se propone el desarrollo de un componente de software que permita la migración de información entre diferentes dispositivos del ambiente. El objetivo de crear este

componente es simplificar y agilizar el desarrollo de aplicaciones de cómputo ubicuo que requieran proveer esta facilidad a los usuarios. Para crear este componente, se utilizó mSALSA que proporciona una librería de clases para dispositivos móviles, basada en la funcionalidad del middleware SALSA. Este middleware permite crear agentes autónomos para ambientes de cómputo ubicuo [6]. Así, el componente de migración consiste de un conjunto de agentes autónomos que interactúan entre sí para permitir la transferencia de información entre dispositivos.

2. Componentes y agentes de software

Dos de los enfoques que se han seguido para desarrollar sistemas complejos son: 1) el paradigma orientado a agentes, en donde los principales elementos del sistema son identificados como agentes que tienen control de su propio comportamiento, y que son capaces de tomar decisiones dependientes del contexto. 2) Utilizar componentes de software como los bloques esenciales para construir el sistema, lo cual es conocido como “componentware” [2]. Un componente de software es una unidad de composición de aplicaciones de software, que posee un conjunto de interfaces y satisface un conjunto de requisitos, y que puede ser desarrollado, adquirido e incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio.

Cada uno de los elementos de construcción utilizados en estos enfoques posee características que lo hacen atractivo para ser utilizado en el desarrollo de sistemas. Por lo anterior, en la creación del componente de migración se utilizaron agentes autónomos los cuales fueron creados con una extensión del middleware SALSA para dispositivos móviles, llamado mSALSA.

2.1. Middleware SALSA

SALSA es un middleware que facilita la construcción y evolución de sistemas de cómputo ubicuo mediante agentes autónomos [6]. Tal como ilustra la figura 1, SALSA proporciona un API (Application Programming Interface) que incluye un conjunto de clases en Java que facilitan la implementación de los agentes. Un Agent Broker que facilita la comunicación entre agentes. Y un servicio de Directorio de Agentes, el cual los agentes utilizan para registrarse o buscar otros agentes a través de sus atributos o por los servicios que ofrecen. La principal desventaja que presenta SALSA es el bajo nivel de granularidad que posee. Puesto que, el alcance del

dominio de la aplicación se limita a la funcionalidad que presenta su conjunto de clases, las cuales para ser implementadas requieren amplio conocimiento por parte del desarrollador. Lo anterior nos ha motivado a aprovechar las características que los agentes SALSA ofrecen [6] para crear un componente de software que permita la fácil implementación e integración de funcionalidad para migrar información entre dispositivos móviles y fijos de un ambiente ubicomp.

2.2. Extendiendo SALSA

Para el desarrollo de este componente en dispositivos móviles haciendo uso de SALSA, se encontró el problema de tener que utilizar funciones nativas del sistema operativo, como por ejemplo el manejo del registro de Windows para generar los menús contextuales de migración de información. Debido a que el API de SALSA en Java no permite implementar lo anterior, se implementó un subconjunto de clases de SALSA desarrollado en el lenguaje de programación C#, al cual se le dio el nombre de mSALSA (micro SALSA).

mSalsa fue desarrollado en C# para tomar ventaja del fácil acceso que tiene hacia las APIs nativas del sistema operativo. A esto le sumamos la lenta inicialización de un sistema desarrollado con Java para PDAs, debido a las incompatibilidades entre los diferentes proveedores de máquinas virtuales de Java, al igual que entre los distintos IDEs y emuladores de PDAs, problemas que al desarrollar con C# no se tienen al utilizar el mismo entorno de desarrollo para todos los dispositivos distintos.

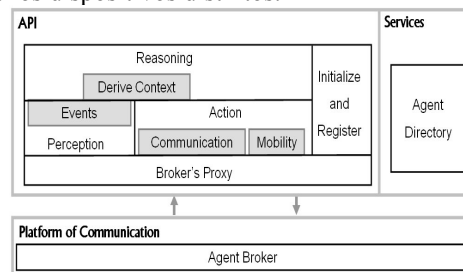


Figura 1. Arquitectura de SALSA

3. Ejemplo de aplicación

Como ejemplo de una aplicación ubicomp desarrollada utilizando mSALSA, se presenta la versión móvil del componente de migración [1]. Dicho componente permite la transferencia de información entre diversos dispositivos que pueden ser heterogéneos. El funcionamiento del componente utiliza como base la funcionalidad provista por los mecanismos y protocolos de comunicación que

proporciona mSALSA, el cual se describe en las siguientes secciones.

3.1. Arquitectura del componente

La figura 2, muestra la arquitectura definida para el componente de migración, la cual contiene agentes construidos con SALSA.

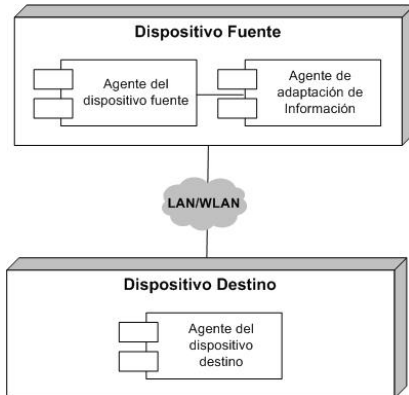


Figura 2. Arquitectura del componente de migración

Los elementos que se presentan en dicha figura son los siguientes:

- *Agente del dispositivo fuente*: este agente representa la información (ya sea el archivo o la referencia a la información) que el usuario desea transferir a otro dispositivo. Así como las características o privilegios (tipo de dispositivo, capacidad, etc.) que posee el dispositivo fuente para permitir la migración de información, y los mecanismos necesarios para transferir la información.
- *Agente de adaptación de información*: este agente se encarga de realizar las modificaciones que debe sufrir la información dadas las características que presenta el dispositivo destino y las especificaciones definidas por el dispositivo fuente.
- *Agente del dispositivo destino*: este agente representa: las características (capacidad, tipos de aplicaciones, etc.) que posee el dispositivo para recibir información; al dispositivo mismo el cual será capaz de aceptar o rechazar la solicitud de transferencia; los mecanismos necesarios para recibir la información.

3.2. Escenario de uso

Un escenario donde se presenta la migración de información es el siguiente:

El Dr. García se encuentra utilizando una pantalla pública para consultar información en el web.

Mientras analiza la información de una página, a través de su pda se le notifica que debe presentarse inmediatamente en urgencias. Dado que, la página web que estaba analizando contiene información sumamente de su interés, decide transferir el url de la página que estaba analizando a su pda. Esto le permitirá continuar posteriormente con el análisis de la información.

En la figura 3, se muestra el flujo de acciones que se realizan para lograr la transferencia de un url de una pantalla pública a una pda. En este caso, la comunicación establecida entre los dispositivos se estableció utilizando los mecanismos de comunicación de mSALSA.

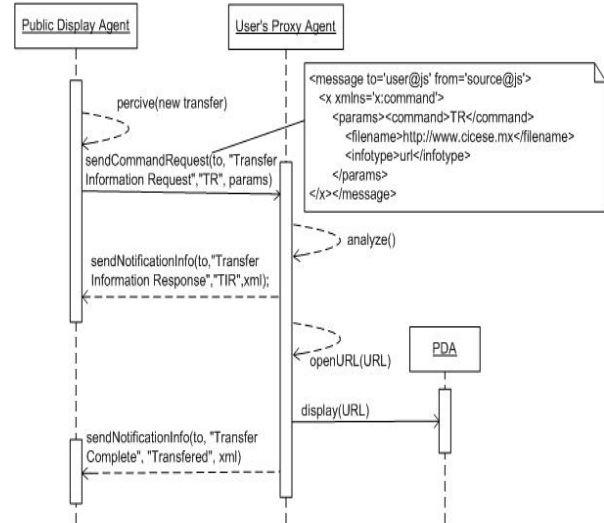


Figura 3. Diagrama de secuencia detallado que muestra la transferencia de una página web de una pantalla pública a una pda

Como puede observarse en la figura, la solicitud de transferencia inicia cuando el agente de la pantalla pública envía un mensaje al agente de la pda a través del método *sendCommandRequest()*. En este mensaje, se especifica la solicitud de transferencia de un url. A partir de dicha solicitud: 1) se determina si se acepta la transferencia; 2) se notifica la aceptación de la transferencia; 3) se despliega el url en la pda; 4) se notifica al agente de la pantalla pública que se realizó la transferencia del url.

3.3. Implementación

La versión actual del prototipo del componente de migración provee soporte para la transferencia de: *archivos* y *URL*. El método de interacción elegido para este prototipo se encuentra implementado como una opción adicional al menú de opciones que proporciona el botón derecho del mouse. El componente consta de dos implementaciones: una para dispositivos móviles

(pda) y otra para escritorio (pc, laptop, pantallas públicas, impresoras). A continuación se describe la transferencia de archivos que se realiza mediante el componente.

3.4. Transferencia de archivos

La transferencia de archivos inicia cuando el usuario mantiene presionada la pluma un par de segundos sobre el archivo que desea transferir hasta que aparecen las opciones. En la figura 4, se observa el resultado de dicha acción.



Figura 4: Pantalla que ilustra la selección del archivo a transferir

Una vez que aparece el menú de opciones, el usuario debe seleccionar la opción de *Transfer To*. Al seleccionar dicha opción, se despliega una nueva pantalla en la cual aparece el listado de todos aquellos posibles dispositivos destinos a los cuales puede transferirse la información. Cuando el usuario elige el dispositivo destino al cual se desea enviar el archivo, se le solicita a éste si desea aceptar la transferencia.

Si la solicitud de transferencia es aceptada por el dispositivo destino, se inicia con la migración de la información. Una vez que la información ha sido transferida se notifica al dispositivo origen que se realizó la transferencia y, en el dispositivo destino se determina la aplicación asociada al tipo de archivo. En el caso de que el dispositivo destino cuente con la aplicación asociada, se abre la aplicación y se despliega el archivo.

5. Conclusión

El crear ambientes de cómputo ubicuo haciendo uso de componentes prefabricados facilita la tarea del

desarrollador ya que esto le permite tomar el componente, introducirlo dentro del contexto de su aplicación y/o ambiente, y adecuar la funcionalidad definiendo las características propias de su aplicación.

Por este motivo, es importante el desarrollo de librerías de clases que nos permitan desarrollar dichos componentes a un nivel más alto de abstracción. Debido a esto, se decidió extender la funcionalidad de SALSA y desarrollar un subconjunto de clases para dispositivos móviles, que sirva como base para el desarrollo de componentes de software para cómputo ubicuo.

Un ejemplo de esto sería el componente de migración presentado en este trabajo. Entre las principales características que provee el componente tenemos: flexibilidad, configuración y reconfiguración, encapsulación, manejo de consciencia del contexto, ejecución independiente, múltiples usos y composición con otros componentes.

6. Referencias

- [1] Amaya, I., Favela, J., Rodríguez, M., Componentes de software para el desarrollo de ambientes de cómputo ubicuo, aceptado para publicación en UCAMI, 2005.
- [2] Bergner, K., Rausch, A., Sihling, M., Vilbig, A., Putting the Parts Together – Concepts, Description Techniques, and Development Process for Componentware, 33th Annual Hawaii International Conference on System Sciences, 2000.
- [3] Greenberg, S., Boyle, M., Laberge, J., PDAs and shared public displays: making personal information public, and public information personal, Personal Technologies, Vol. 3, No.1, pp. 54-64, Marzo 1999.
- [4] Johanson, B., Ponnekanti, S., Sengupta, C., Fox, A., Multibrowsing: moving web content across multiple displays, Proc. 3rd international conference on Ubiquitous Computing, Atlanta, Georgia, USA, pp: 346-353, 2001.
- [5] Konomi, S., Müller-Tomfelde, C., Streit, N., Passage: physical transportation of digital information in cooperative buildings, Proc. of the 2nd International Workshop on Cooperative Buildings (CoBuild'99), LNCS 1670, Heidelberg, Germany, Springer, pp. 45-54, 1999.
- [6] Rodríguez M., Favela, J., Preciado, A., and Vizcaino, A., Agent-based Ambient Intelligence for Healthcare, accepted for publication in AI Communications, 2005.