



UNIVERSIDAD PONTIFICIA DE SALAMANCA

Campus de Madrid

Facultad de Informática

DOCTORADO EN INGENIERÍA INFORMÁTICA

Programa en Ingeniería del Software

BIENIO 2003-2005

ASIGNATURA:

Agentes Móviles y Multiagentes: Tendencias y Metodologías

TRABAJO DE INVESTIGACIÓN:

Análisis Crítico de la Situación actual de la tecnología de agentes en el desarrollo de sistemas multiagentes para dispositivos móviles.

PROFESOR:

Dr. D. Juan Manuel Corchado

AUTOR:

César Parejas Llanovarcad

Madrid, Febrero de 2005

ÍNDICE

INTRODUCCIÓN.....	3
TECNOLOGÍA DE AGENTES.....	4
Plataformas de agentes	4
Lenguajes de programación de plataformas de agentes	5
METODOLOGÍAS DE DESARROLLO DE SISTEMAS MULTIAGENTE	7
Análisis y Diseño Orientado a agentes	7
Técnica de Modelado de Agentes para sistemas de agentes BDI.....	7
Método basado en escenario para sistemas MultiAgentes (MASB)	7
Metodología MAS-CommonKADS	8
Metodología CoMoMAS.....	8
EXTENSIONES DE UML PARA EL MODELADO DE AGENTES	9
Aplicación de diagramas de secuencia a Agentes	9
Aplicación de diagramas de colaboración a Agentes	10
Aplicación de diagramas de actividades a Agentes.....	10
ANÁLISIS DE LA SITUACIÓN ACTUAL DE DESARROLLO DE SISTEMAS MULTIAGENTES PARA DISPOSITIVOS MÓVILES	11
Arquitecturas de agentes para dispositivos móviles	11
Tecnología J2ME-Java 2 Micro Edition.....	11
Tecnología de agentes y J2ME.....	14
Proyectos de Investigación	15
Agentes móviles en SAIPE: Sistema de Acceso a Información Personal desde Entornos con conectividad limitada	15
Sistema multiagente para la gestión de agendas personales.....	17
Otros Proyectos Europeos sobre Agentes.....	19
TENDENCIAS Y LINEAS DE INVESTIGACIÓN DE LOS SISTEMAS MULTIAGENTES PARA DISPOSITIVOS MÓVILES	19
CONCLUSIONES.....	21
BIBLIOGRAFÍA Y REFERENCIAS	22

INTRODUCCIÓN

Los avances de las tecnologías de la información y las comunicaciones, y el auge actual de las comunicaciones inalámbricas ha dado lugar a la utilización cada vez más de dispositivos móviles con acceso a redes tales como PDA, Pocket Pc, Ordenadores Portátiles, Teléfonos Móviles, etc.

Por otro lado, en los últimos años el desarrollo Orientado a Agentes de ha convertido en un nuevo paradigma de Ingeniería del Software. El concepto de agente constituye una potente herramienta de abstracción en el desarrollo de software, que facilita la construcción de sistemas distribuidos, inteligentes y robustos.

Actualmente dentro del paradigma de la programación orientada a agentes están cobrando especial relevancia los agentes móviles, esto es, programas que pueden parar en cualquier momento, trasladarse a otros dispositivos y continuar su ejecución en el punto en que se suspendió lo cual, unido a las características propias de los agentes, permite llevar los cálculos al dispositivo en el que se encuentran los datos disminuyendo así el tránsito de información por una red. En este sentido, el lenguaje Java ha sido el que más éxito ha tenido para la implementación de infraestructuras de agentes.

El presente trabajo de investigación, pretende analizar el panorama actual de las arquitecturas de desarrollo de sistemas multiagentes orientados a los dispositivos móviles, además de presentar una visión de la orientación actual de los proyectos enfocados en esta área. Para concluir, analizaremos las tendencias que se avizoran en un futuro próximo.

TECNOLOGÍA DE AGENTES

La tecnología de agentes ha tenido un especial impacto en los últimos años gracias a su aplicación en la computación distribuida. La ventaja que proporciona respecto a los modelos clásicos cliente/servidor es que permite realizar las mismas operaciones pero con la ventaja de que son asíncronas y que además no se precisan conexiones permanentes para la ejecución de tareas, puesto que el agente que migra hacia otro sistema además de llevar los datos necesarios para realizar la operación, conserva la información de estado del proceso.

El concepto de agente ha sido muy discutido y aunque existen varias definiciones, la más aceptada es la que define a los agentes por sus características: movilidad, autonomía, inteligencia, comunicación, cooperación y coordinación. Un agente es aquel que posee una o varias de estas características.

La solución de problemas complejos no se lleva a cabo por un único agente, sino que es necesario un conjunto de agentes que interactúan entre ellos para la consecución de los objetivos del sistema. La utilización de varios agentes representa de forma natural la descentralización del sistema y enfatiza la comunicación para gestionar las dependencias entre ellos.

Plataformas de agentes

Para introducir los conceptos relacionados con plataformas de agentes seguiremos la definición del estándar FIPA, que es el estándar más ampliamente reconocido y extendido en tecnología de agentes.

FIPA define una plataforma de agentes como la infraestructura, tanto hardware como software, que precisan los agentes para ser desarrollados y usados. Esta plataforma contendría agentes especiales que proporcionarían servicios a los agentes que se encuentren en la plataforma. FIPA define tres servicios obligatorios en una plataforma de agentes:

- Agent Management System (AMS), que gestiona la plataforma y los agentes, ofreciendo, entre otros, los siguientes servicios:
 - Control de ciclo de vida de un agente, siguiendo su diagrama de estados (ver figura 1).

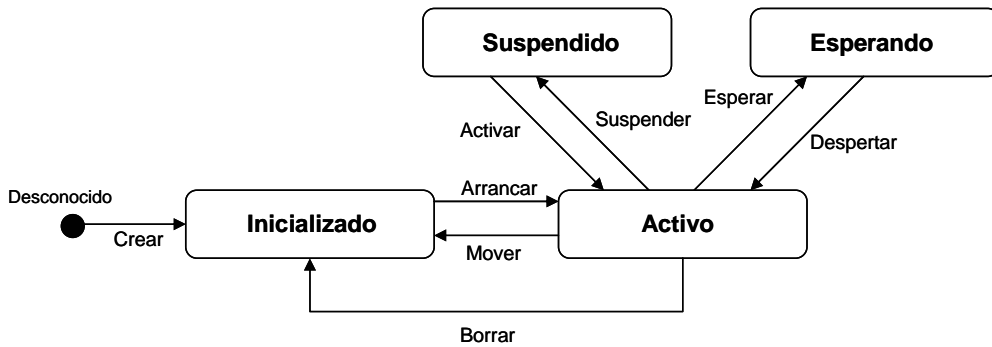


Figura 1. Ciclo de estado de los agentes

- Registro de los agentes.
 - Control de movilidad de los agentes.
 - Gestión de recursos compartidos.
 - Gestión del canal de comunicación.
 - Servicio de páginas blancas para localizar agentes por su nombre.
- Director Facility (DF), que complementa al servicio de nombres, facilitando la localización de agentes según los servicios que ofrecen.
 - Agent Communication Channel (ACC), que gestiona el envío de mensajes entre agentes de una plataforma y entre agentes de distintas plataformas.

Lenguajes de programación de plataformas de agentes

Una parte importante del trabajo realizado en el desarrollo de plataformas de agentes ha sido proporcionar o adaptar lenguajes de programación para implementar estas plataformas. Se han utilizado lenguajes de propósito más general como pueden ser Java o adaptaciones de otros, como las versiones realizadas sobre Tcl: Safe-Tcl, Agent Tcl, TACOMA y en algunos casos, se han definido lenguajes específicos para este propósito: Telescript, M0, Tycoon.

La característica principal que deben proporcionar estos lenguajes es permitir la movilidad de una unidad de ejecución, que consiste en una parte de código, que nos proporciona la descripción estática del comportamiento de un programa, y el estado de la unidad de ejecución. El estado contiene por una parte, lo que se denomina espacio de datos, que son todos los recursos accesibles desde todas las rutinas activas y por otra, lo que se denomina estado de ejecución, que es información de control, como el valor del contador del programa y el estado de la pila, que nos permite retomar la ejecución después de la migración. Cada uno de estos componentes pueden moverse independientemente.

Los diferentes lenguajes han aportado soluciones diferentes para la movilidad de estos componentes, agrupándose en dos clases, por una parte los lenguajes que soportan *strong mobility* y los que se soporta *weak mobility*:

- **Strong mobility:** tienen la capacidad de migrar código y estado, abarcando tanto el espacio de datos como el estado de ejecución.
- **Weak mobility:** tienen la capacidad de migrar código y el espacio de datos del estado del código, pero no el estado de ejecución.

Aunque existen lenguajes que proporcionan *Strong mobility*, como son Telescript, TACOMA, Ara y D'Agents. La mayoría de los lenguajes soportan *weak mobility*, entre ellos Java, que a pesar de esta limitación ha sido el lenguaje en el que más plataformas de agentes se han desarrollado, debido a las siguientes ventajas aportadas por el lenguaje:

- Independencia de la plataforma.
- Ejecución segura.
- Carga dinámica de clases.
- Programación multithread.
- Serialización de objetos.
- Reflexión.

A pesar de todas estas ventajas, el desarrollo de plataformas de agentes con Java tiene ciertas limitaciones, una de las más importante es la que comentamos anteriormente, que sólo soporta *weak mobility*, pero además: no proporciona un soporte adecuado para el control de recursos, ni proporciona control de referencias, todos los métodos públicos de un objeto Java son accesibles desde cualquier otro objeto que tenga una referencia a él.

METODOLOGÍAS DE DESARROLLO DE SISTEMAS MULTIAGENTE

El nuevo enfoque del paradigma orientado a Agentes, ha suscitado diferentes aproximaciones, definiciones y metodologías de desarrollo, que se basan en este nuevo modelamiento del “mundo”.

La mayoría de las metodologías orientadas a agentes tienen un sustento teórico basado en metodologías basadas en objetos. Pero a su vez existen algunas que tienen sustentos teóricos de Ingeniería de Conocimiento. A continuación se hará una síntesis de algunas de algunas metodologías existentes:

Análisis y Diseño Orientado a agentes

Esta técnica (Burmeister 1996) define tres modelos para el análisis de sistemas basados en agentes: el modelo agente, que contiene a los agentes y su estructura interna (creencias, planes y metas); el modelo organizacional, que describe las relaciones entre los agentes (herencia y roles en la organización); y el modelo de cooperación, que describe las interacciones entre los agentes.

Técnica de Modelado de Agentes para sistemas de agentes BDI

BDI define dos niveles principales (interno y externo) para los agentes BDI (Belief, Desire and Intention).

El punto de vista externo consiste de una descomposición del sistema en agentes y la definición de sus interacciones. Esto se ve a través de dos modelos: el modelo de agente, para describir las relaciones de jerarquías entre los agentes; y el modelo de interacción, para describir las responsabilidades, servicios e interacciones entre los agentes y sistemas externos.

El punto de vista interno se lleva a cabo a través de tres modelos: el modelo de creencias, que describe las creencias sobre el ambiente; el modelo de metas, que describe las metas y eventos que un agente puede adoptar o responder; y el modelo de planes, que describe los planes que un agente puede utilizar para conseguir sus metas.

Método basado en escenario para sistemas MultiAgentes (MASB)

Este método (Moulin y Cloutier 1994) es aplicado para sistemas multiagente en el campo de trabajo cooperativo.

La fase de análisis consiste de las siguientes actividades: descripción de escenarios, descripción funcional de roles, modelado conceptual del mundo y datos, y modelo de interacción sistema-usuario.

La fase de diseño consiste de: arquitectura MAS y descripción de escenario, modelado de objetos, modelado de agentes, modelado de conversaciones y validación total del sistema de diseño.

Metodología MAS-CommonKADS

Esta metodología [15] comienza con una fase de conceptualización que es una fase informal que se utiliza para recolectar los requerimientos de usuario y se obtiene una primera descripción del sistema desde el punto de vista del usuario.

La metodología define los siguientes modelos:

1. **Modelo Agente:** describe las características principales de los agentes, incluyendo la capacidad de razonamiento, servicios, metas, etc.
2. **Modelo de Tarea:** describe las tareas (metas) que ejecuta un agente y una descomposición de las mismas.
3. **Modelo Expertise:** describe el conocimiento necesario que necesitan los agentes para llevar a cabo las tareas.
4. **Modelo de Coordinación:** describe las conversaciones entre agentes, esto es, sus interacciones, protocolos y capacidades requeridas.
5. **Modelo de Comunicación:** detalla las interacciones humano-agente a través del software, y los factores humanos para desarrollar estas interfaces.
6. **Modelo de Diseño:** recolecta los modelos anteriores y los subdivide en diseño de aplicación, diseño arquitectónico y diseño de plataforma.

Metodología CoMoMAS

Esta es una extensión a la metodología CommonKADS (Schreiber 1994) para el modelado MAS. Se definen los siguientes modelos:

1. **Modelo de Agente:** define la arquitectura de los agentes y el conocimiento de los mismos. Se clasifican como conocimientos sociales, cooperativos, de control, cognitivos y reactivos.
2. **Modelo Expertise:** describe la competencia reactiva y cognoscitiva del agente.
3. **Modelos de Tarea:** describe la descomposición de tareas y detalles si la tarea fue resuelta por un usuario o agente.
4. **Modelo de Cooperación:** describe la cooperación entre agentes.
5. **Modelo de Sistema:** define los aspectos organizacionales de la sociedad del agente conjuntamente con los aspectos arquitectónicos.
6. **Modelos de Diseño:** recoge los modelos previos con el objeto de hacerlos operacionales junto con los requerimientos no funcionales.

EXTENSIONES DE UML PARA EL MODELADO DE AGENTES

Aplicación de diagramas de secuencia a Agentes

La figura 8 muestra los elementos básicos para la comunicación entre agentes. El rectángulo puede representar tanto agentes de forma individual como también un conjunto de agentes agrupados. Por ejemplo un agente individual puede etiquetarse *Juan/Cliente*. Aquí Juan es una instancia de un agente que está representando el rol de Cliente. Para indicar que Juan es una Persona – independientemente del rol que esté interpretando– Juan puede expresarse como *Juan:Persona*. La manera básica de etiquetar un agente es *nombre-agente/rol:clase*. La sintaxis mencionada antes ya es parte de UML pero con la diferencia que UML indica el nombre de un objeto en lugar de un agente. Se muestra un ejemplo en la Figura 2.

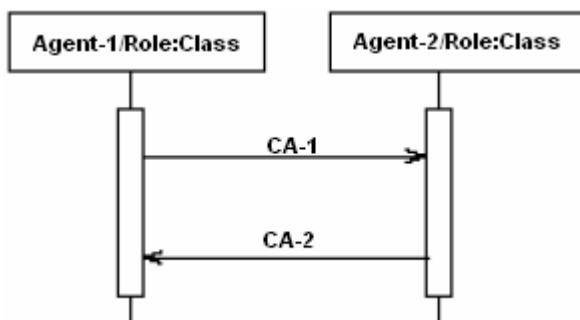


Figura 2. Diagrama de Secuencia

Otra recomendación importante de UML es la posibilidad de considerar el soporte para hilos concurrente de interacción. La siguiente figura muestra tres maneras distintas de expresar hilos múltiples. La primera figura indica que todos los hilos CA-1,.....,CA-n se envían de manera concurrente.

La segunda figura incluye una caja de decisión que decidirá que CAs (cero o más) se enviarán. Si se envía más de una CA la comunicación es concurrente. La figura 3 muestra una or excluyente, de manera que una sola CA se enviará.

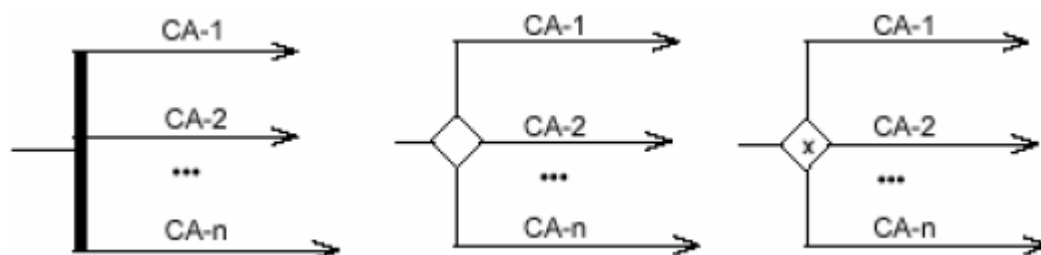


Figura 3. Diagrama de Secuencia con OR Excluyente

Aplicación de diagramas de colaboración a Agentes

Una de las diferencias existentes con el diagrama de secuencia es que en este diagrama los agentes pueden ser situados en cualquier lugar del gráfico, dando una mayor facilidad al momento de hacer el diagrama. En cuanto al significado del diagrama es el mismo que el diagrama de secuencia, teniendo en cuenta que la ejecución o llamadas de los mensajes están dados en las etiquetas de los mismos por el número que tienen.

Aplicación de diagramas de actividades a Agentes

Los protocolos de comunicación de agentes algunas veces requieren de especificaciones con una semántica muy clara en lo que se refiere al procesamiento de hilos de ejecución. Los diagramas de actividades expresan las operaciones y los eventos que disparan a los hilos de ejecución.

Proveen una representación gráfica que hace posible la visualización de procesos de una manera simple, simplificando de esta manera el diseño y la comunicación del modelo de comportamiento. Por otra parte, se puede representar cualquier tipo de procesamiento concurrente y asíncrono. También pueden representar comunicaciones simultáneas.

ANÁLISIS DE LA SITUACIÓN ACTUAL DE DESARROLLO DE SISTEMAS MULTIAGENTES PARA DISPOSITIVOS MÓVILES

Arquitecturas de agentes para dispositivos móviles

Con la aparición de versiones de Java para pequeños dispositivos, J2ME para PDA's o teléfonos móviles, y considerando que el lenguaje de programación en el que más plataformas de agentes móviles se han desarrollado es Java, las posibilidades de desarrollar plataformas de agentes en estos dispositivos se presenta alcanzable.

En esta sección se realizará previamente un análisis de las limitaciones, tanto de lenguaje como de recursos, a las que nos enfrentamos a la hora de migrar tecnología de agentes a dispositivos limitados con J2ME y a continuación, se analizarán las propuestas e iniciativas existentes en la literatura relacionadas con este tema.

Tecnología J2ME-Java 2 Micro Edition

En 1999, Sun Microsystems anuncia la aparición de Java 2 Micro Edition con el propósito de permitir que aplicaciones Java se ejecuten en dispositivos con potencia de procesamiento limitada, como teléfonos móviles, pagers, palm pilots, set-top boxes, y otros. Una solución que responde a la amplia difusión que están teniendo estos dispositivos en los últimos años y a la demanda de usuarios y proveedores de servicios de recibir/ofrecer nuevas aplicaciones para aumentar las funcionalidades que aportan estos pequeños dispositivos.

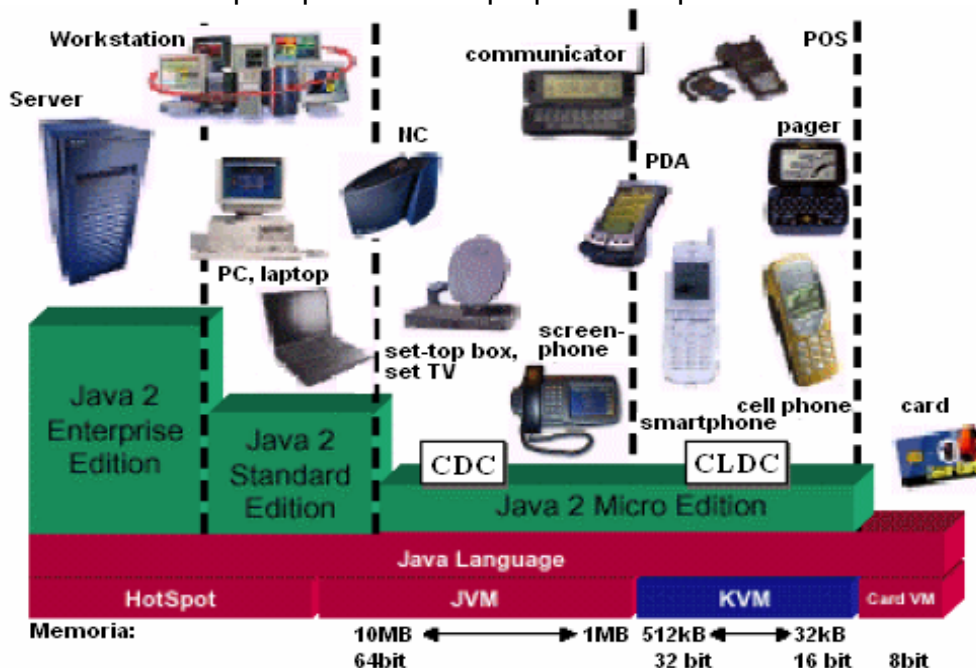


Figura 4. Visión por capas de la tecnología Java

J2ME en la actualidad abarca dos categorías de dispositivos, por un parte los que se denominan fijos, que poseen conexiones a la red fija y tienen una capacidad de almacenamiento del orden de los 2 a 16 megaBytes de memoria. Por ejemplo, set-top boxes, Internet TV y sistemas de navegación de automóvil. Y por otra parte, los dispositivos denominados móviles, que tienen capacidades de almacenamiento limitadas del orden de los 128 kiloBytes, con microprocesadores del 16 o 32 bit RISC/CISC y que se comunican a través de conexiones inalámbricas. Dentro de esta categoría están los teléfonos móviles, palm pilots y pagers.

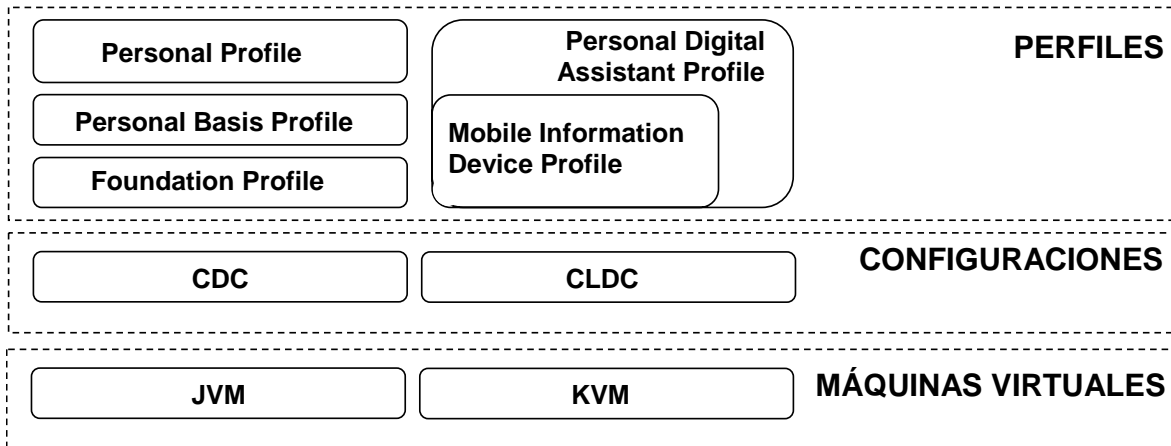


Figura 5. Arquitectura J2ME

Una de las principales ventajas de J2ME es que es una arquitectura modular (figura 5) que se adapta a las limitaciones de los diferentes dispositivos en las que se quiere integrar. Se definen tres capas:

- **Máquina virtual.** En la actualidad J2ME soporta dos máquinas virtuales: la Java Virtual Machina que se emplea en ediciones J2SE y en J2EE para los dispositivos con procesadores de 32 bit, y la KVM para arquitecturas de 16/32 bits pero con capacidades de almacenamiento limitado.
- **Configuraciones.** Definen las características del núcleo de la plataforma.
 - Características soportadas del lenguaje Java
 - Modelo de Seguridad
 - Capacidades de interconexión

Se establecen una serie de bibliotecas Java que están disponibles para un conjunto de dispositivos, con similares capacidades de procesamiento y memoria. J2ME soporta varias configuraciones, en la actualidad existen dos estandarizadas:

- **Connected, Limited Device Configuration (CLDC),** que engloba en general a dispositivos personales móviles.

- **Connected Device Configuration (CDC)**, que engloba en general a dispositivos fijos. Por motivos de compatibilidad es un superconjunto de CLDC.

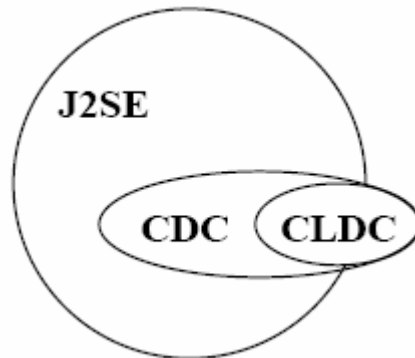


Figura 6. Relación entre CLDC y CDC y la J2SE

Ambas configuraciones tienen clases comunes con la J2SE, que permite la compatibilidad, pero poseen además clases específicas para los tipos de dispositivos para los que se definieron.

- **Perfiles.** Definen un conjunto de API's que pueden emplearse para desarrollar aplicaciones para una familia particular de dispositivos. El principal objetivo en la definición de un perfil es garantizar la interoperabilidad de las aplicaciones entre un conjunto de dispositivos que soportan el mismo perfil. Un mismo dispositivo puede soportar diferentes perfiles. Los perfiles se desarrollan sobre una determinada configuración. Así sobre CLDC se ha estandarizado el Mobile Information Device Profile (MIDP) para teléfonos móviles y pagers y se encuentra en proceso de de estandarización el PDA Profile, para asistentes personales. Sobre CDC se están estandarizando el RMI Profile, Foundation Profile, Personal Profile entre otros.

Para el inicio del desarrollo de un sistema de agentes en una plataforma J2ME, es necesario analizar las limitaciones que presentan sus configuraciones. En este sentido, analizamos las limitaciones de los dispositivos que se engloban en la configuración CLDC:

- **Limitaciones del lenguaje Java:** no soporta tipos de datos de coma flotante (float o double), no soporta finalización de instancias de clases y tiene limitaciones en el manejo de errores.
- **Limitaciones de la máquina virtual:**
 - No soporta Java Native Interface (JNI).
 - No soporta cargadores de clase definidos por el usuario.
 - No soporta reflexión, y por lo tanto, ni serialización de objetos, ni soporte a RMI, ni otras características avanzadas de Java (JVM Debugging Interface, JVM Profile Interface)

- No soporta grupos de threads ni daemon threads, las operaciones de arranque y parada de threads sólo se pueden aplicar individualmente.

La mayoría de estas limitaciones se deben a las propias limitaciones de procesamiento y memoria de los dispositivos y a razones de seguridad motivadas porque J2ME/CLDC no soporta el modelo completo de seguridad de J2SE.

En el futuro se pretenden suplir algunas de estas limitaciones, entre ellas la sincronización de threads y el cargador dinámico de clases.

Si observamos las características que convertían a Java en un buen lenguaje para desarrollar plataformas de agentes, y las comparamos con las restricciones de J2ME, vemos que gran parte de éstas han desaparecido, o se han visto limitadas por motivos de seguridad, en concreto, aquellas que nos permitían implementar movilidad de objetos (carga dinámica de clases, serialización y reflexión).

Tecnología de agentes y J2ME

Existen varias propuestas que pretenden involucrar a dispositivos limitados J2ME en plataformas de agentes móviles, en este apartado haremos referencia a aquellas más importantes.

- **JADE-LEAP.** El proyecto LEAP engloba un consorcio de compañías entre las que se encuentran, entre otras, Motorola, British Telecom y Siemens. El objetivo del proyecto es el desarrollo de una plataforma de agentes en Java conforme al estándar FIPA (Foundation for Intelligent Physical Agents) que pueda operar tanto en dispositivos limitados (teléfonos móviles, PDA, pagers) con J2ME, como en PC's con J2SE. Para ello, han diseñado una arquitectura modular, con una parte obligatoria, común a todos los tipos de dispositivos, y otra opcional; y mediante un instalador se podría componer la plataforma según las limitaciones del dispositivo en el que se instale.
- **Monash University.** En Monash University se están realizando proyectos relacionados con el desarrollo de plataformas de agentes para dispositivos móviles empleando J2ME, en particular se han realizado desarrollos para PDA. Su propuesta se basa en incluir dentro de la KVM una plataforma de agentes que habían desarrollado para entorno PC, con algunas restricciones, obteniendo de esta forma unas mejores prestaciones, pero obligando a que los dispositivos tengan esta KVM reconstruida.
- **School of Computer Science of Carleton University.** En School of Computer Science of Carleton University también se están realizando desarrollos orientados a la utilización de tecnología de agentes para aplicaciones en entornos móviles, pero en su propuesta la plataforma de agentes reside en un dispositivo no limitado denominado Agent Gateway que sirve de mediador

entre el dispositivo inalámbrico y los recursos de la red. La justificación de esta propuesta es que los dispositivos móviles tienen recursos limitados y que en la actualidad la especificación de la J2ME/CLDC no tiene funcionalidades básicas para la realización de plataformas de agentes, como es la serialización de objetos o la carga dinámica de clases.

Proyectos de Investigación

Agentes móviles en SAIPE: Sistema de Acceso a Información Personal desde Entornos con conectividad limitada

SAIPE es un sistema basado en agentes móviles que facilita al usuario el acceso a información personal cuando usa dispositivos móviles y una conexión de red inestable, como puede ser las redes inalámbricas basadas en GPRS.

SAIPE permite acceder a cuentas de correo electrónico para buscar mensajes de correo y leer sólo los que interesen, sin que para ello se tengan que obtener todos los demás mensajes que puede haber en dicha cuenta. Además permite buscar documentos usando alguno de los sistemas de búsqueda registrados en el sistema y después descargarlos si así se desea, desplazando los procesos mas pesados del Dispositivo de Usuario desde el que se ejecuta la aplicación a un dispositivo distinto con mayor capacidad de cálculo, denominado Dispositivo Servidor. Ver Figura 7

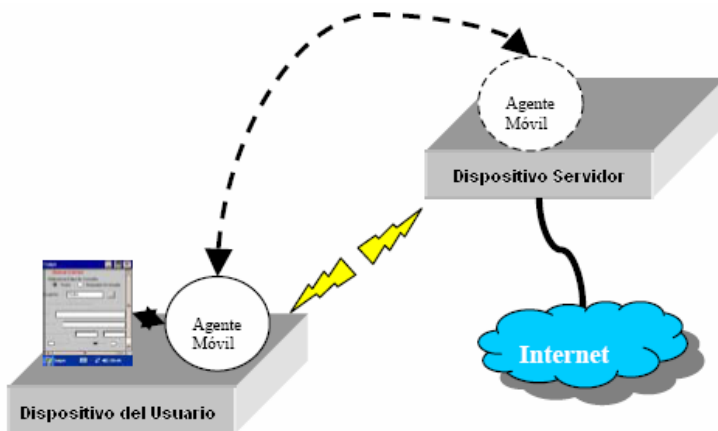


Figura 7. Distribución de dispositivos

Cuando se ordena a la aplicación hacer una búsqueda de documentos o de mensajes de correo electrónico, leer mensajes de correo o descargar documentos, la interfaz le ordena a uno de los agentes que forman parte del sistema que realice dicha acción. Este agente (agente móvil) cuando recibe la orden se desplaza al Dispositivo Servidor para que realicen los cálculos, en donde hace todas las operaciones necesarias para dar respuesta a la petición. Una vez obtenida la información regresa al origen con la información obtenida. Esta información se muestra al usuario a través de la interfaz del sistema.

Para el desarrollo del sistema, utilizaron la plataforma de agentes móviles JADE-LEAP.

Arquitectura. El sistema sigue una arquitectura basada en agentes móviles de la que forman parte los agentes Interfaz, BuscarDocumentos y Cartero. Estos agentes se inician en el dispositivo del Usuario cuando se arranca el sistema.

El agente Interfaz es un agente estático que se encarga de gestionar las comunicaciones con el usuario (operador humano) mediante una interfaz gráfica con la que interactúa. Los otros dos agentes son agentes móviles que se trasladan a un dispositivo distinto del usuario (Dispositivo Servidor) para ejecutar en éste las acciones que les que ordene el agente Interfaz. Una vez se desplazan al Dispositivo Servidor ejecutan las acciones que les fueron solicitadas haciendo uso de la conexión a Internet de banda ancha ofrecida como servicio a agentes móviles así como la utilización de los servicios necesarios. Una vez obtienen la respuesta a la acción regresan al Dispositivo del Usuario y le comunican al agente interfaz esa respuesta.

El agente móvil Cartero realiza todas las funciones de manejo de correo electrónico mientras que el agente móvil BuscadorDocumentos lleva a cabo las operaciones sobre documentos (por ejemplo, búsqueda o descarga de documentos, etc.).

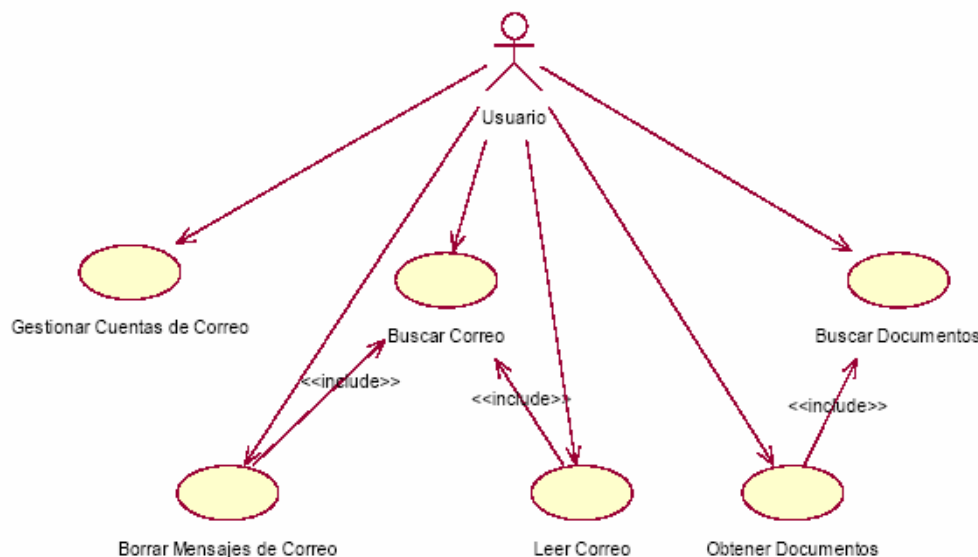


Figura 8. Diagrama de Casos de Uso de SAIPE

TAgentsP: un perfil sobre CLDC para agentes móviles

El proyecto TAgentsP (Traveling Agents Profile) tiene por objetivo implementar una plataforma de agentes en dispositivos limitados CLDC/KVM, que permita que los agentes se puedan mover entre estos dispositivos, sin necesidad de involucrar a dispositivos no limitados, para realizar sus tareas.

Sus desarrollos actuales se centran en complementar el MIDP para construir un perfil sobre el CLDC que proporcione la funcionalidad básica de un plataforma de agentes móviles, Cuando hablamos de funcionalidad básica nos referimos a dotar a J2ME de aquellas características que tenía Java como lenguaje de desarrollo de plataformas de agentes y que J2ME no posee, en concreto las que nos permiten weak mobility:

- Carga dinámica de clases.
- Serialización de objetos.

Una vez proporcionados los servicios básicos, y para conseguir este objetivo de que los agentes se puedan mover entre dispositivos limitados, requirieron el envío de clases que precisan los agentes para su ejecución, para ello implementaron un reducido servidor HTTP en cada dispositivo. Esta implementación también permitió la comunicación a nivel agente.

Sistema multiagente para la gestión de agendas personales

Proyecto Europeo IST-2000-30045 COLLABORATOR [1]. En este sistema, los agentes se encargan de obtener un perfil de los usuarios del sistema analizando sus agendas personales y las reuniones que los usuarios han establecido en ellas. Una vez definido el perfil de cada usuario, los agentes son capaces de negociar entre ellos nuevas reuniones teniendo en cuenta las preferencias de los usuarios.

En el sistema COLLABORATOR cada usuario tiene una agenda personal y asociada a ella un agente encargado de analizar su contenido. Este agente personal se encarga de establecer el criterio seguido por el usuario a la hora de fijar las reuniones en su agenda personal. Para ello el agente clasifica las reuniones y almacena la tendencia que tiene el usuario a fijar un tipo de reunión en determinada fecha. Basándose en estos perfiles los agentes personales de los usuarios pueden negociar nuevas reuniones sin que los usuarios tengan que intervenir. Una vez fijada la reunión el sistema informa a cada usuario del resultado de la negociación para que estos puedan confirmar o rechazar su participación. Esta información sirve de realimentación a los agentes personales para corregir el perfil que poseen del usuario.

Detección de perfiles de usuario

El perfil de cada usuario se compone de información encaminada a la categorización de reuniones e información referente a las preferencias temporales del usuario para fijar reuniones en función de la categoría a la que pertenece. El usuario define el conjunto de categorías especificando propiedades de las reuniones tales como asistentes, palabras clave que definen el contenido de la reunión y lugar de la reunión. Por su parte el sistema genera automáticamente el conjunto de preferencias temporales observando los intervalos horarios en los que un determinado usuario fija, acepta o rechaza reuniones de una determinada clase. Esta información es expresada mediante hechos que conforman la base de

conocimiento del motor de inferencias del agente personal. Un usuario puede visualizar en todo momento las preferencias temporales inducidas por su agente personal, pudiendo añadir nuevas restricciones o bien modificar las ya existentes.

Este perfil debe adaptarse automáticamente a los cambios que un usuario experimenta a lo largo del tiempo a la hora de planificar/clasificar nuevas reuniones. Esto lo consiguen monitorizando en todo momento las interacciones del usuario con el sistema. En el caso de que el usuario detecte como errónea la clasificación de una reunión, el sistema refuerza negativamente los términos comunes a la reunión y a la categoría seleccionada por el agente y positivamente esos mismos términos en la categoría seleccionada por el usuario. Además cada vez que una nueva reunión es introducida por el usuario, el agente verifica si en base a sus preferencias temporales sería capaz de planificar la reunión en esa misma fecha o en su entorno. En función del resultado obtenido, el grado de confianza en los hechos almacenados en la base de conocimiento del agente y que le han llevado a realizar dicha planificación es reforzado positiva o negativamente, eliminándolos si esta confianza está por debajo de un umbral.

Negociación de reuniones

El sistema COLLABORATOR define el concepto de sesión. Una sesión se compone de una serie de aplicaciones orientadas al trabajo en grupo y un conjunto de participantes. Para gestionar el ciclo de vida de estas sesiones se definen los agentes de sesión. Su principal tarea consiste en hacer de mediador entre los diferentes agentes de usuario a la hora de negociar la fecha en la que va a tener lugar una determinada sesión. El administrador de una sesión propone un conjunto de fechas en los cuales la sesión debe tener lugar así como un conjunto de invitados, los conocimientos deseables en dichos invitados, una descripción textual del tema de la reunión, etc. Toda esta información es transmitida desde el agente de sesión a cada uno de los agentes personales de los invitados. Éstos agentes responden al agente de sesión con información relativa a las preferencias temporales y disponibilidad de los usuarios para cada una de esas fechas. Para obtener dichas preferencias los agentes personales consultan el perfil de su usuario: clasificando primero la reunión en una de las categorías definidas para después obtener el grado de disposición del usuario a asistir a dicha reunión en función de la hora así como su disponibilidad. Con toda esta información el agente de sesión determina una fecha para la reunión maximizando una función que depende del número de asistentes y de sus preferencias. Este resultado se comunica a los agentes personales de todos los invitados y estos responden confirmando o rechazando la asistencia de sus usuarios. Si algunos de los agentes personales rechaza su asistencia de forma inesperada para el agente de sesión se busca un nuevo máximo comparándolo con el anterior. Si el nuevo máximo es menor que el anterior el sistema se queda con el resultado obtenido inicialmente. En caso contrario se repite todo el proceso hasta agotar todas las posibilidades.

Además de mediar en la negociación los agentes de sesión se encargan de informar a los agentes de usuario de cambios en el estado de la sesión tales como cancelación, inicio, un usuario decide no asistir, etc., con el fin de garantizar la sincronización de todas las agendas personales de los usuarios.

Otros Proyectos Europeos sobre Agentes

- **AGENTLIGHT.** Propuesta de una plataforma para Agentes Ligeros.
- **AMBIESENSE.** Sistemas de información ambiental personalizado para dispositivos móviles.
- **ANIMATED AGENTS.** Integración de Agentes animados en un entorno de aprendizaje virtual para personas con discapacidad.
- **AGENT ACADEMY.** Presenta una plataforma de minería de datos para agentes inteligentes orientados al entrenamiento.
- **AGENCOM.** AgenCom proporciona una arquitectura para desarrollar, desplegar, y gestionar aplicaciones e-business. La arquitectura se basa en un entorno distribuido de n capas donde un número de capas de la lógica de la aplicación y servicios de negocio son realizados por agentes que se comunican entre sí a través de la red.
- **AGENTCITIES.** AgentCities es una iniciativa mundial orientada a ayudar en la comprensión del potencial de la investigación de aplicaciones basadas en agentes. El objetivo es construir una red mundial de plataformas de agentes basadas en el estándar FIPA. Cada plataforma será apoyada por diversas organizaciones capaces de tener acceso a los servicios de forma compartida.
- **ALIVE.** El objetivo del proyecto es desarrollar una arquitectura y un sistema de herramientas, genéricas y dependientes de la aplicación, para el mejoramiento de espacios narrativos. Esto se logrará mediante la prueba de dos aspectos de este objetivo: tele presencia (inclusión de objetos verdaderos en el mundo virtual) y realidad aumentada (inclusión de objetos artificiales en el mundo real).

TENDENCIAS Y LINEAS DE INVESTIGACIÓN DE LOS SISTEMAS MULTIAGENTES PARA DISPOSITIVOS MÓVILES

Algunas de las tendencias de esta tecnología en sistemas de telefonía móvil serían:

- Por una parte, en tareas de gestión de red, siguiendo una tendencia ya explorada en redes fijas, pero que tiene mayor interés en redes móviles debido a que las propias características de los agentes móviles se adaptan a las limitaciones de los sistemas inalámbricos.
- Por otra, en la realización del VHE (Virtual Home Environment), que permitirá la personalización y portabilidad de los servicios de los usuarios

independientemente de la red que le da servicio y del terminal que empleen en el acceso. Asociamos la implementación del VHE con un agente móvil, que permitirá configurar el servicio para adaptarse a las preferencias del usuario y a las características del terminal, y además será el encargado de crear el propio perfil de usuario analizando su comportamiento y su posición.

En cuanto a la seguridad de las plataformas de agentes, se plantean las siguientes líneas de investigación:

- Mecanismo de distribución de claves públicas entre servidores
- Protección total de los datos del agente y ejecución privada
- Restricción del consumo de HD y RAM por agente
- Intercambio de mensaje e información entre agentes de forma segura

Otra línea de investigación a destacar, es la Portabilidad (los agentes móviles deben ser capaces de moverse en redes heterogéneas), Interoperabilidad (Es necesario que los agentes de un sistema puedan interactuar e intercambiar información con los de otro sistema o bien con objetos de otras tecnologías) e implementación de la tecnología de agentes:

- Interoperabilidad con otros sistemas de agentes móviles
- Integración de los agentes móviles en los navegadores
- Integración de la tecnología con otras tecnologías como CORBA
- Una implementación más eficiente

CONCLUSIONES

- Hemos analizado algunas de las metodologías y herramientas para el desarrollo de agentes móviles como base en el desarrollo de un sistema distribuido donde la conectividad de los usuarios puede estar condicionada por factores externos. De igual forma, hemos resumido algunos de los proyectos de investigación más interesantes que actualmente se están desarrollando con la tecnología de agentes móviles.
- La implementación de una infraestructura de agentes en una red de dispositivos móviles presenta una complejidad importante debido a las limitaciones que imponen los terminales, que todavía poseen una capacidad de procesamiento y almacenamiento reducida. Con la aparición de versiones de Java para pequeños dispositivos, como Java Card para tarjetas inteligentes y j2ME para PDA's o teléfonos móviles, y considerando que el lenguaje de programación en el que más plataformas de agentes móviles se han desarrollado es Java, las posibilidades de desarrollar plataformas de agentes en dispositivos móviles se presenta alcanzable.
- Existen un conjunto de problemas abiertos para la implantación de esta tecnología. Por una parte el tema de la seguridad, que es un tema crítico en todas las implementaciones de sistemas de agentes móviles y más aún en el caso de sistemas de telefonía móvil, en los que se debe garantizar itinerancias de servicios y por lo tanto de agentes, a través de diferentes redes de diferentes operadores. El tema es crucial y para su implementación se deberían establecer políticas de seguridad entre organismos de diferentes países.
- La realización del presente trabajo resultó una experiencia interesante en la comprensión de la situación actual de las tecnologías de agentes móviles, en particular en el desarrollo de sistemas multiagentes para dispositivos móviles. Es mi deseo, como objetivo particular, el seguir profundizando en la investigación de esta tecnología informática, desde el punto de vista práctico.

BIBLIOGRAFÍA Y REFERENCIAS

- EUROPEAN CO-ORDINATION ACTION FOR AGENT-BASED COMPUTING
<http://www.agentlink.org/resources/agentprojects-db.html>
- Bellifemine, F., et al, 2002. Jade Programmer's Guide (JADE 3.1).
<http://sharon.csel.it/projects/jade/>
- COLLABORATOR (COLLABORative fRAMework for remoTe and mObile useRs) IST-2000-30045. Available at <http://www.ist-COLLABORATOR.net>
- Lange, D. 1998. Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley.
- DeLoach, S., Wood, M. June 2000. Multiagent Systems Engineering: The Analysis Phase.
- Aproximación Metodológica para el desarrollo de Software Orientado a Sistemas
FACULTAD DE INFORMATICA UNIVERSIDAD POLITECNICA DE MADRID
Multiagente. Doctor Francisco Javier Segovia. Junio 2003
- CAEPIA-2003. Agentes Inteligentes en Tercer Milenio.
Organizado por: Antonio Moreno, Universidad Rovira i Virgili, Tarragona. Nov. 2003
- Plataforma de Agentes de Terminales de Telefonía Móvil.
Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid.
M^a Celeste Campo Vázquez, Carlos García Rubio.
- FIPA. FIPA Agent Managment Specification, March 2002. <http://www.fipa.org/>
- Agentes Móviles en computación ubicua
Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid.
M^a Celeste Campo Vázquez
- Agentes Móviles en SAIPE
Departamento de Informática de la Universidad de Vigo.
Jose A. Barcala, Pedro Cuesta, Alma Gómez.