Nómadas (Col)
ISSN: 0121-7550
nomadas@ucentral.edu.co
Universidad Central
Colombia

Martínez Barrera, Crisman
AGENTES DE SOFTWARE MÓVILES
Nómadas (Col), núm. 15, octubre, 2001, pp. 281-294
Universidad Central
Bogotá, Colombia

AGENTES DE SOFTWARE MÓVILES

Crisman Martínez Barrera*

Los agentes móviles son programas de software inteligentes que realizan un objetivo que involucran desarrollos soportados en técnicas de Inteligencia Artificial, los cuales pretenden facilitar la interoperabilidad de sistemas. Este artículo define las disciplinas, plataformas y herramientas necesarias para el desarrollo de agentes móviles, sus características principales y las arquitecturas predominantes de éstas; presenta además una evaluación de sus perspectivas futuras.

Mobile agents are intelligent software programs that can obtain an objective that involucrates developments supported in Artificial Intelligence techniques. These pretend to facilitate the interoperability of systems. This article defines disciplines, platforms and tools necessary for the development of mobile agents, their principal characteristics and the predominant architectures of these. A final evaluation and future perspectives are offered.

* Ingeniero de Sistemas. Profesor de la Escuela de Ingeniería de la Universidad Central de Bogotá. Candidato a Magíster en Teleinformática de la Universidad Distrital Francisco José de Caldas, Bogotá. Analista de sistemas y consultor en tecnologías de punta. E-mail: mcrisman@visto.com

En la actualidad, existe una gran variedad de productos de software disponible para los usuarios, la cual proporciona servicios en múltiples dominios; pero estas aplicaciones no son de gran utilidad si trabajan de forma aislada. Existe una demanda creciente que pide la interoperabilidad entre programas para intercambiar información y servicios para resolver problemas.

La interoperabilidad de sistemas encuentra su dificultad en la heterogeneidad, estas aplicaciones pueden estar desarrolladas en diferentes herramientas y sobre diferentes plataformas, dando como resultado diferentes interfaces.

El software basado en agentes pretende facilitar la interoperabilidad de sistemas[13]. Los agentes de software móviles proveen un bus software (conceptual) independiente de: *Quién* los diseñó, *Qué* plataforma lo soporta, *Cómo* se programaron (lenguaje), y *Dónde* se ejecutan[14].

El software basado en agentes móviles pretende crear, gestionar, mantener, comunicar e inter-operar los sistemas que están desarrollados en diferentes herramientas y plataformas. Un agente de software móvil tiene las características de autonomía, sociabilidad, reactividad, proactividad, inteligencia y movilidad, como se explica mas adelante.

Este artículo pretende introducir al lector en el mundo de los sistemas multiagentes mediante la presentación de las características y estándares que rigen los agentes y que se han convertido en un punto de referencia inevitable para el futuro desarrollo de las tecnologías emergentes en Internet.

Agentes de Software Móvil

Los agentes móviles son programas de software inteligentes que realizan un objetivo y pueden estar construidos con herramientas de desarrollo estructuradas, orientadas a objetos y/o concurrentes. Normalmente involucran desarrollos soportados en técnicas de Inteligencia Artificial.

Los agentes actúan de manera autónoma; están programados para que perciban y aprendan de su entorno a través de sensores, razonan sobre lo aprendido; eligen una o varias soluciones para resolver problemas o consultar requerimientos automáticos o humanos; viajan sobre la plataforma de comunicaciones de una red LAN o WAN; sirven para automatizar tareas que se ejecutan repetidamente en una red; se programan una única vez y posteriormente no requieren manipulación humana.

Los agentes de software móviles aprenden de manera inteligente del entorno de red donde se ejecutan y son transparentes en: quién los diseñó, qué plataforma los soporta, cómo se programaron (lenguaje de desarrollo) y dónde se ejecutan.

Para desarrollar un agente de software móvil se requiere:

- Una plataforma de gestión de agentes: (Aglets, MOA, ARA, Grasshopper, Odyssey, JATLite).
- Una(s) herramienta(s) de desarrollo de software: estructurada (C), orientado a objetos (C++) o concurrente (Java).
- Un entorno de red.

Es importante que para su desarrollo se tengan en cuenta las normas de estandarización de las organizaciones internacionales. Algunas de ellas, las más importantes se mencionan en este artículo.

En este artículo se presentan algunas definiciones de agente, sus principales características, su arquitectura, las organizaciones de estandarización internacional, así como las ventajas y desventajas de los agentes de software móviles.

Definiciones de Agente

Actualmente hay tres disciplinas informáticas fundamentales en el desarrollo y definición de agentes[1].

1. Inteligencia artificial.
2. Programación orientada a objetos y programación concurrente.
3. Diseño de interfaces hombre-máquina.

La definición formal de agente de software es de las grandes ausentes cuando se trabaja con la tecnología de agentes. Cada autor frecuentemente da su propia definición. Indicaré, a modo de ejemplo, las definiciones más usadas:

Según el Diccionario de la Lengua Española[2]: "Agente: Del latín agens, entis, p. a. de agere, hacer.

1. adj. Que obra o tiene virtud de obrar.
2. m. Persona o cosa que produce un efecto.
3. Persona que obra con poder de otro ...".

Según el Object Management Group [3]: "Un agente es un programa de ordenador que actúa autónomamente en nombre de una persona u organización".

Según Russell y Norving en su artículo "Artificial Intelligence: a Modern Approach"[4]: "Un agente puede verse como aquello que percibe su entorno a través de sensores y que actúa sobre él, mediante efectores".

Según Hayes-Roth[5]: "Los agentes inteligentes realizan continuamente tres funciones: percepción de las condiciones dinámicas de un entorno, acción (que afecta a dichas condiciones) y razonamiento (para interpretar percepciones, resolver problemas, hacer inferencias y determinar acciones)".

Según I.B.M. Aglets[5]: "Los agentes inteligentes son entidades programadas que llevan a cabo una serie de operaciones en nombre de un usuario o de otro programa, con algún grado de independencia o autonomía, empleando algún conocimiento o representación de los objetivos o deseos del usuario".

Desde mi punto de vista, los agentes móviles corresponden a un conjunto de instrucciones inteligentes que realizan un objetivo y pueden estar construidos con herramientas de desarrollo estructuradas, orientadas a objetos y/o concurrentes.

Este agente actúa de manera autónoma, está programado para que perciba y aprenda de su entorno a través de sensores, razona sobre lo aprendido, elige una o varias soluciones para resolver problemas o consultar requerimientos automáticos o humanos, y puede o no viajar sobre la plataforma de comunicaciones de una red LAN o WAN.

Para viajar sobre la arquitectura de gestión (Figura 1), el agente realiza los siguientes pasos:

1. Inicia la ejecución en el transmisor.
2. Realiza o no una serie de eventos en el transmisor.

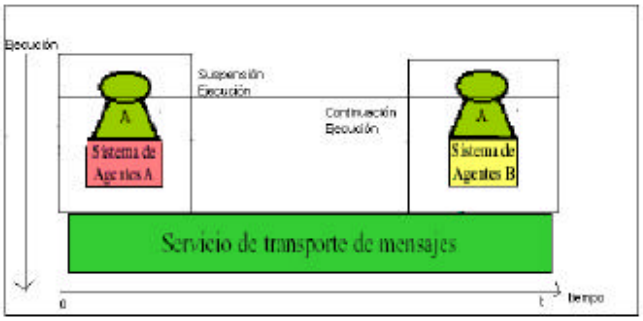


Figura 1. Ciclo de Ejecución de un Agente móvil.

3. Suspende la ejecución en el transmisor.
4. Transfiere el código por la red (canal de comunicaciones).
5. Despierta en el receptor.
6. Continúa su ejecución (en el receptor) en el punto donde se había suspendido (en el transmisor).
7. Realiza o no una serie de eventos en el receptor.
8. Finaliza o suspende la ejecución para devolver el requerimiento al transmisor.

Características de un Agente

Las características de un sistema de agentes de software móvil según el estándar FIPA (Foundation for Intelligent Physical Agents) son[6]:

Autonomía

La autonomía es una de las características más importantes del concepto de agente. El software tradicional suele ejecutarse en entornos interactivos de tal forma que responde a ordenes directas del usuario, es decir, el usuario tiene que decir paso a paso qué es lo que se tiene que hacer.

La idea de los agentes consiste en crear programas informáticos que tengan una serie de objetivos y posean unos conocimientos del mundo, de tal forma que partiendo de sus conocimientos, sean capaces de aproximarse lo más posible a sus objetivos sin necesidad de que ningún usuario los guíe paso a paso hacia ellos.

Sociabilidad

Cuando se habla de agente no se suele pensar en una única entidad que se ejecuta de forma aislada. Más bien se

piensa en sistemas complejos (multi-agente) en los que una serie de agentes colaboran entre sí para llevar a cabo una tarea. Este modelo denominado tradicionalmente como "divide y vencerás" presupone que los agentes son capaces de interactuar entre sí y al mismo tiempo, hacerlo con entidades externas al propio sistema como es el caso del usuario.

Reactividad

A pesar de su autonomía, un agente debe ser capaz de percibir estímulos externos tanto para actuar de acuerdo a su entorno cambiante como para poder "conocer" en todo momento cómo es el "mundo" que le rodea. Es decir, que el agente debe tener estímulos y actuar de acuerdo con ellos. Estos estímulos afectarán a las acciones realizadas por él para alcanzar sus objetivos.

Pro-actividad

Es una de las consecuencias de la autonomía de un agente. Éste es capaz de elegir, en cada momento, cuáles son las acciones a realizar para alcanzar sus objetivos. Es decir, un agente no solo actúa en función de los estímulos que recibe desde el exterior, sino que puede ejecutar acciones como resultado de sus propias decisiones.

Inteligencia

Un agente es inteligente si es racional, coherente, adaptable y móvil, en mayor o menor medida. Podemos decir que será más inteligente cuanto más desarrolladas tenga estas características.

1. Racionalidad

La racionalidad es una característica propia del ser humano. Un agente se considera racional cuando tiene conocimientos de su entorno, objetivos deseables y reglas que determinan cómo alcanzar los objetivos a partir del conocimiento que se tiene del medio.

Esta característica le permite a un agente tomar decisiones sin la intervención humana. De momento en problemas muy simples se está modelando la racionalidad propia del hombre.

2. Coherencia

El conocimiento que un agente tiene de su mundo se almacena en una base de datos de conocimiento interna al propio agente. Todo este conocimiento debe guardar un alto grado de coherencia para que el comportamiento del agente sea el esperado.

3. Adaptabilidad

El aprendizaje o adaptabilidad es una de las características más complejas que se le puede pedir a una entidad de software inteligente. Un agente aprende cuando es capaz de aumentar su base de conocimiento y su base de reglas a partir de las percepciones que recibe del entorno y de sus comportamientos anteriores a la hora de resolver problemas.

Ésta es una característica bastante deseable debido a que el entorno de un agente suele ser dinámico en el tiempo y es necesario que se adapte al mismo.

4. Movilidad

Ésta es una característica opcional que pueden poseer los agentes. Un agente móvil es aquel que se puede mover físicamente por los nodos de una red para llevar a cabo sus tareas. El objetivo de la movilidad puede ser una mejor distribución de la carga de procesamiento, una mejor repartición de recursos, o una lógica distribuida.

Arquitectura de los Agentes Móviles

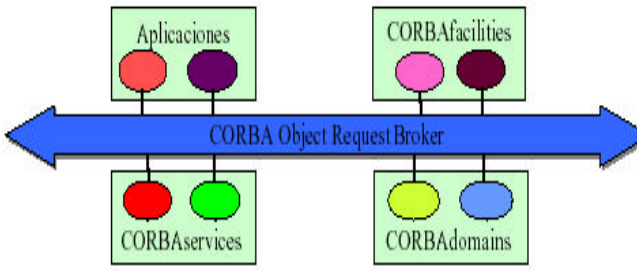
Uno de los objetivos más importantes de la tecnología de agentes móviles es la interoperabilidad entre diferentes sistemas de agentes. Para facilitar dicha interoperabilidad es necesario especificar de una forma estándar acciones como la transferencia de agentes y la transferencia de clases y operaciones de control de los agentes. Cuando los sistemas de agentes de origen y destino son similares (mismo lenguaje de implementación), la estandarización de estas acciones asegura la interoperabilidad. Sin embargo, si los sistemas son diferentes sólo una mínima interoperabilidad puede conseguirse, ya que el código que se transfiriere a un sistema con un lenguaje diferente no puede ejecutarse[17].

Según las organizaciones de estandarización internacional:

1. OMG (Object Management Group)[7]. Mobile Agent System Interoperability Facilities (MASIF)

Dentro de la arquitectura de CORBA se contemplan los agentes de software móvil (ASM) como objetos CORBA que tienen la posibilidad de moverse, ejecutarse autónoma y asincrónicamente en sistemas de ejecución seguros, y facilitar la creación de sistemas de agentes móviles.

La arquitectura CORBA (*Common Object Request Broker Architecture*) es una especificación estándar de un modelo o protocolo de comunicación entre objetos distribuidos. Está basada en un gestor de peticiones a objetos comunes y permite la interoperabilidad entre aplicaciones en máquinas remotas en un entorno distribuido completamente heterogéneo como se observa:

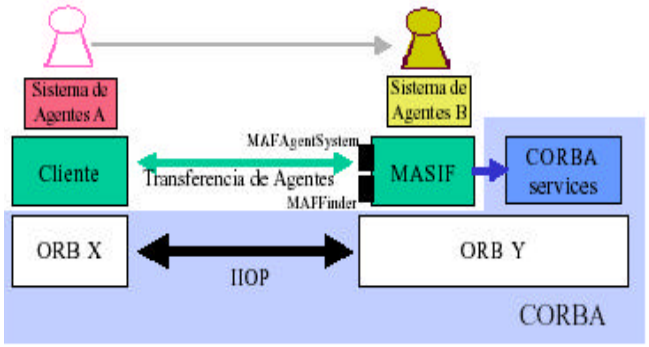


Arquitectura CORBA

Por otro lado, la implementación de una aplicación u objeto depende del lenguaje de desarrollo; sin embargo, para poder interoperar con otras aplicaciones u objetos, sólo es necesario disponer de sus interfaces. En el desarrollo de la implementación, la interoperabilidad entre las aplicaciones u objetos, se realiza mediante referencias a los objetos implicados y utilizando sus interfaces. La arquitectura software que da soporte a esta interoperabilidad mediante el uso de referencias es otro componente fundamental en CORBA llamado ORB (*Object Request Broker*). ORB es un objeto que gestiona las peticiones a los objetos comunes, su implementación depende del lenguaje de desarrollo utilizado aunque ofrece una funcionalidad común en todas las plataformas[17].

La interoperabilidad entre ORB's resuelve el problema de cómo un cliente con un ORB invoca las operaciones de un objeto en un servidor con otro ORB.

Los clientes y servidores con ORB's pueden encontrase en la misma máquina o distribuidos, también pueden estar implementados con el mismo lenguaje de programación o con diferentes (C++, Java, ADA, etc.). La implementación del ORB y de su interfaz con los clientes y servidores (así como la implementación de éstos) es dependiente del lenguaje empleado. Es necesaria una interfaz y protocolo de comunicación entre ORB´s independiente de su implementación. Para ello CORBA proporciona los puentes (*bridges*) y el protocolo GIOP:



Estandarización de sistemas de agentes en CORBA

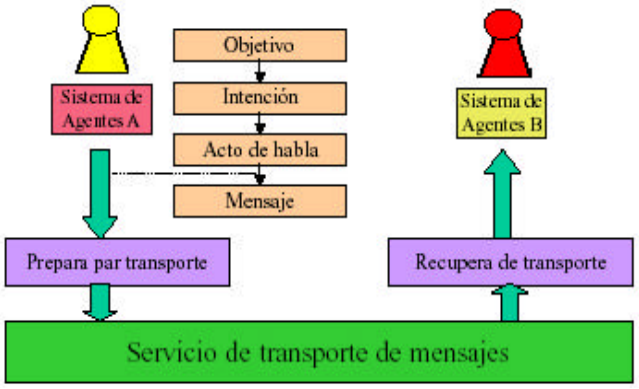
2. ARPA:

KSE (Knowledge Sharing Effort)
Knowledge Querying and Manipulation Language (KQML)
Knowledge Interchange Format (KIF)[8]

El KSE (*Knowledge Sharing Effort*) es un consorcio centrado en el desarrollo de normas que facilitan la repartición y reutilización de bases de conocimiento y sistemas basados en el conocimiento.

Para KSE se requiere Sintaxis (KIF), Semántica (Ontolingua) y Pragmática (KQML), todos corresponden al lenguaje de comunicación común que soporta la capacidad de compartir conocimiento entre agentes:

Como se observa el nivel de contenido (mensaje) hace referencia a la información que realmente se está intentando comunicar. Para que dos o más agentes puedan establecer un diálogo, es necesario que establezcan un lenguaje común para la representación de la infor-



Estandarización de sistemas de agentes según KSE

mación o que empleen algún tipo de notación que pueda ser fácilmente traducible al lenguaje que cada uno de ellos utiliza. Para poder interpretar los mensajes, una vez establecido un lenguaje común traducible, es necesario que utilicen un vocabulario común: una ontología. Ésta consiste en un conjunto de términos los cuales se presenta el conocimiento, es como un diccionario de clases, relaciones, funciones, objetos y constantes empleadas en el área de conocimiento en la que se está trabajando.

En el nivel de intención, la misma información del nivel de contenido puede comunicarse con diferentes intenciones. Básicamente el nivel de intención de un mensaje es el tipo de dicho mensaje. La construcción de un conjunto de mensajes, lo suficientemente expresivos como para que puedan ser utilizados por cualquier tipo de sistema de agentes cooperantes no es una labor sencilla. Investigadores en el tema han desarrollado un conjunto de primitivas de comunicación (*performatives*) basadas en la teoría para el modelado de la comunicación humana *speech act theory* que describen el conjunto de primitivas necesarias para la coordinación entre agentes.

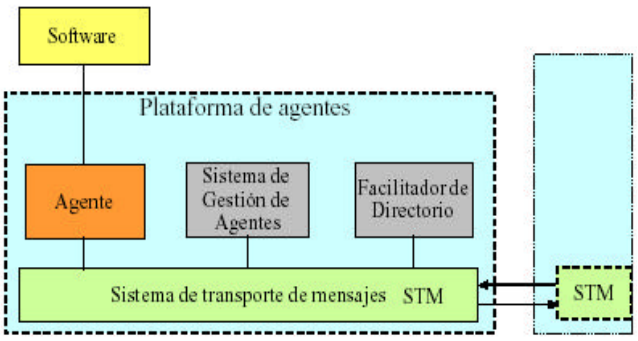
Los tipos de mensajes pueden ser clasificados en tres grupos: asertivos, directivos y declarativos. Un mensaje es asertivo cuando comunica un hecho, por ejemplo las primitivas "envía" o "recibe"; directivo cuando indica un comando o acción, por ejemplo una pregunta o sugestión: "dime todo lo que sepas acerca de ...", o "te informo que ..."; y declarativo cuando envía información acerca de sus capacidades, como por ejemplo "sé responder interrogantes con el siguiente formato ...".

El nivel de coordinación (acto de habla), se refiere a la forma en que se establecen los diálogos entre los agentes. ¿Qué convenios deben establecer los agentes para intercambiar mensajes de un modo eficiente?. Existen diferentes protocolos de coordinación: *Contrat Net Protocol AgentTalk, COOL* Por ejemplo, el protocolo de coordinación COOL es un lenguaje para la coordinación que modeliza la conversación entre los agentes mediante una máquina de estados finitos. Cada estado representa el punto de la conversación en el que nos encontramos. El paso de un estado a otro viene determinado por la llegada del mensaje correcto. De este modo se puede establecer la conducta de los agentes frente a la llegada de los mensajes[17].

3. FIPA (Foundation for Intelligent Physical Agents)

Recoge todas las miradas vistas que se tienen de un sistema de agentes (gestión, seguridad, movilidad, comunicación, etc.). De tal forma que para cualquiera de las mismas todos los esfuerzos se orientan en una única dirección[17]. Además reúne las especificaciones de arquitectura, infraestructura y aplicaciones:

El Agent Management proporciona la normativa del entorno donde los agentes FIPA se crean y operan. Establece el modelo lógico de referencia para la gestión de agentes (creación, registro, localización, comunicación, migración y terminación de los agentes). Este modelo presenta un conjunto de capacidades lógicas y no implica ninguna configuración física, simplemente deja los detalles de implementación a elección del equipo de desarrollo.



Estandarización de agentes en FIPA

El Agent Management o modelo de referencia para la gestión de agentes, está formado por los siguientes componentes lógicos:

Agente: es el componente básico y principal del modelo. Combina una o más capacidades de servicio dentro de un entorno de ejecución integrado y unificado que proporciona servicios de comunicación y acceso al software externo y a los usuarios.

Un agente tiene que tener uno o más dueños. Además debe disponer de una identidad propia proporcionada por un identificador global y único GUID (*Globally Unique Identifier*) denominado nombre del agente. Un agente puede registrarse con un número de direcciones en las cuales puede ser contactado.

Agent Platform (AP): proporciona la infraestructura física y lógica necesaria para que los agentes puedan ejecutarse. Una plataforma de agentes está constituida por el hardware (puede haber varios computadores), el sistema operativo, el software de comunicaciones y el software de agentes.

Directory Facilitator (DF): componente que siempre tiene que aparecer en cualquier plataforma de agentes FIPA. Es un agente que proporciona un servicio de páginas amarillas a los demás agentes. Un agente puede utilizar DF para registrar sus servicios o para encontrar los servicios ofrecidos por otros agentes.

Agent Management System (AMS): componente que siempre tiene que aparecer en cualquier plataforma de agentes FIPA. Existe uno por plataforma. Es un agente de gestión que controla el estado y el acceso a la plataforma. También proporciona un servicio de páginas blancas que permite la localización de agentes a partir de sus nombres.

Agent Communication Channel (ACC): todos los agentes tienen acceso al menos a un ACC. El ACC es el canal de comunicación por defecto entre agentes de diferentes plataformas. Tiene que soportar el protocolo de comunicación para interoperabilidad IIOP.

Internal Platform Message Transport (IPMT): método de intercambio de mensajes dentro de la misma plataforma. Depende de la implementación.

4. Agent Society

Arquitectura y protocolos de comunicación genéricos [10].

Destaca los agentes que están desarrollados en plataformas, protocolos de transferencia, lenguajes de programación y lenguajes de comunicación diferentes.

Según esta figura, el éxito de *Common Agent Plataform* y *Simple Agent Transport Protocol* dependerá de su compatibilidad con las especificaciones de MASIF y FIPA evolucionando hacia la interoperabilidad y los estándares abiertos.

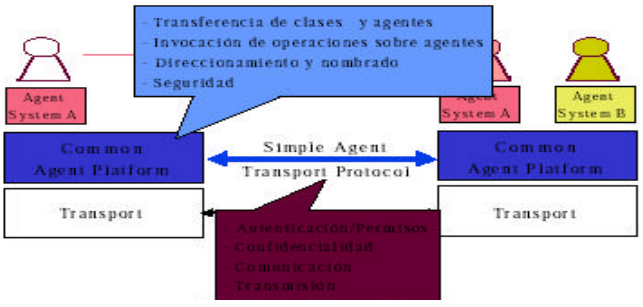
Movilidad de los Agentes

Una vez conocidas las diferentes arquitecturas de los agentes de software móviles de cada una de las compañías de estandarización internacional, es necesario conocer también los aspectos de comunicación e intercambio de información entre ellos.

La interoperabilidad entre estos sistemas es muy importante debido a que si dos sistemas de distinto origen no pueden intercambiar agentes, la movilidad se queda como una capacidad no aprovechable.

Para lograr la movilidad de los agentes se puede trabajar con dos soluciones[6]:

1. *Migración*: este modelo es más complejo ya que requiere la transferencia de códigos y datos del agente a la plataforma remota. Cuando un agente va a migrar se suspende, se transfiere por la red y



Arquitectura de plataforma común de agentes

se despierta en el otro extremo siguiendo su ejecución en el mismo punto donde la había dejado.

2. *Clonación*: en este caso se crea una copia del agente en el nodo remoto y éste se encarga de hacer el trabajo solicitado por su clon en la máquina origen. En este caso no hay una transferencia efectiva de código y datos, sino solo un conjunto de órdenes que el clon debe realizar.

Modelo de Movilidad Simple:

El AMS (*Agent Management System*) es el responsable de realizar toda la gestión necesaria. El agente solicita a su AMS la transferencia y ésta se ocupa de llevarla a cabo (protocolo de migración simple).

Protocolos de movilidad simple: el agente delega en un protocolo de alto nivel su operación de movilidad a una determinada plataforma como se ilustra a continuación. El protocolo soporta una única acción (*move*); en este caso, la plataforma en la que se ejecuta el agente tiene que implementar dicho protocolo necesario para la operación de migración[17].

Las ventajas del protocolo simple son:

1. Reduce la complejidad del desarrollo del agente debido a que la movilidad la proporciona la plataforma[17].

2. Está orientado hacia sistemas existentes de agentes móviles (como MASIF) y facilita la implementación en plataformas ya existentes mediante el lenguaje de comunicación de agentes de FIPA (ACL).

3. Cuenta con número de interacciones remotas reducido.

Modelo de Movilidad Complejo:

El agente es el responsable de realizar toda la operatividad que le permite desplazarse a la plataforma remota (protocolo de migración complejo):

Protocolo de movilidad complejo: el agente dirige el protocolo necesario para su operación de movilidad y no delega la responsabilidad en la plataforma. Primero el agente mueve su código y estado a la plataforma destino y una vez el nuevo agente ha sido creado con éxito le transfiere su identidad y autoridad. Además, este protocolo también permite que el agente informe a la plataforma en la cual fue creado (*home agent platform HAP*) y a otras plataformas de agentes (*agents platforms AP's*) que se ha movido a una nueva localización.

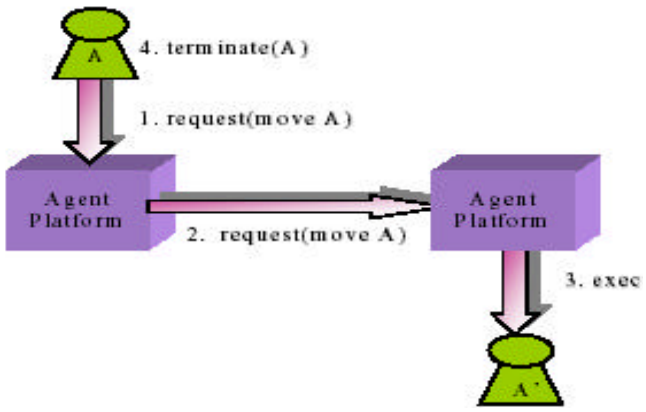
Las ventajas del protocolo complejo son:

1. Reduce la implementación de la plataforma debido a que son los agentes quienes proporcionan la movilidad.

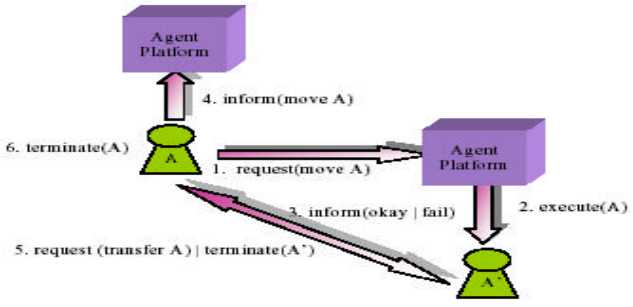
2. Capacita al agente para controlar la operación de movilidad.

3. Constituye una forma más segura de movilidad.

Cuando un agente inicie con la acción *move* una operación de movilidad (tal como migración,



Modelo de Movilidad simple



Modelo de Movilidad Complejo

clonación o invocación), el agente tendrá que indicar el protocolo de movilidad que ha de usarse en la operación. Con esta información, el AMS implicado determinará los pasos a realizar para completar la operación, lo cual puede requerir el uso de otras acciones como *transfer*[17].

Integración de agentes MASIF - FIPA

La movilidad entre los estandares MASIF y FIPA se logra mediante los puntos de acceso de cada uno de los elementos de las dos arquitecturas.

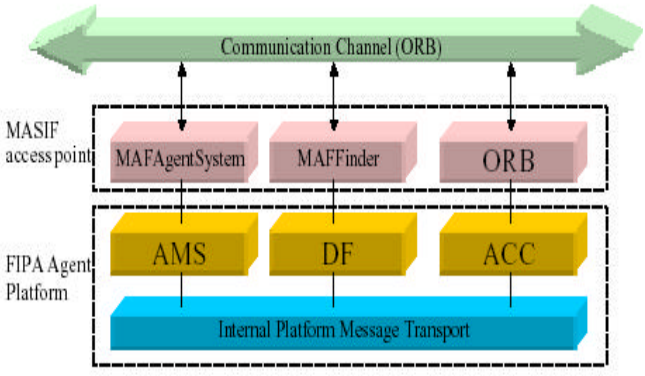
La intención del estándar MASIF es ofrecer interoperabilidad entre plataformas de agentes móviles con implementaciones diferentes. Es posible el acceso a una plataforma de agentes MASIF mediante dos interfaces estándar que son especificadas mediante el Lenguaje de Definición de Interfaces (IDL) de OMG, conforme a la siguiente gráfica. Estas interfaces denominadas *MAFAgentSystem* y *MAFFinder* proporcionan las operaciones fundamentales para la gestión y transporte de agentes, así como el registro de componentes. También constituyen el medio de acceso al sistema de agentes[17].

Pueden encontrarse algunas características comunes entre una plataforma MASIF y una FIPA relativas a la funcionalidad especificada:

El FIPA Agent Management System (AMS) puede compararse con el sistema de agentes MASIF representado por la interfaz *MAFAgentSystem*. Ambos tienen la responsabilidad de la gestión de los agentes (creación, terminación, suspensión, autentificación, migración, etc.).

El FIPA Directory Facilitator (DF) es similar al componente MASIF para los registros representado por la interfaz *MAFFinder*. La tarea de estas entidades es mantener la información registrada de los agentes en un entorno distribuido.

El equivalente del *FIPA Agent Communication Channel* (ACC) es el *Object Request Broker* (ORB) en el contexto MASIF. Estas entidades se encargan de la transferencia de mensajes en un entorno distribuido de agentes.



Movilidad de los agentes entre MASIF y FIPA

FIPA y MASIF proporcionan sus especificaciones independientes de la implementación.

El estándar FIPA abarca toda la funcionalidad necesaria para la ejecución y soporte de agentes móviles mediante un lenguaje de descripción de contenidos y acciones de alto nivel (*Agent Communication Language* ACL). El ACL permite la especificación de operaciones y la utilización de protocolos de comunicación de alto nivel.

El estándar MASIF no constituye una base completa para el desarrollo de nuevos sistemas sino que simplemente abarca una mínima funcionalidad para así poder adaptarse a las plataformas de agentes ya existentes. La funcionalidad de MASIF se establece mediante interfaces IDL y no utiliza ningún lenguaje de alto nivel sobre los métodos IDL, sino que directamente se corresponde con métodos en la implementación de los objetos.

Estos dos estándares pueden ser combinados para unificarse en un único marco de agentes móviles. Un camino prometedor parece ser la integración en la especificación IDL de MASIF, de operaciones IDL correspondientes a las definidas en FIPA mediante la transferencia de mensajes ACL. Para realizar una plataforma de agentes compatible FIPA y MASIF, existen tres posibilidades:

1. Las interfaces existentes en MASIF (*MAFAgentSystem* y *MAFFinder*) tienen que incorporar nuevas operaciones que permitan el acceso a la plataforma FIPA.

2. Las operaciones de las interfases de MASIF deben ser modificadas adaptándolas a los requerimientos de las especificaciones FIPA.

3. Nuevas interfaces deben ser añadidas.

La especificación FIPA también podría añadir algunos métodos definidos en el estándar MASIF. Tendrían que ser métodos que tuvieran una estructura de parámetros simple y que pudieran representarse sin usar un lenguaje de contenidos de alto nivel.

Ventajas de los Agentes de Software Móviles

- La consolidación de la tecnología de agentes es la mejor forma de resolución de problemas, mucho mejor que la utilizada hasta el momento. Sirve incluso para la resolución de problemas que no habían podido resolverse por no disponer de la tecnología adecuada o porque la utilización de la existente implicaba un costo excesivo.

- Reducen el trabajo del usuario y del programador de aplicaciones. El agente puede adaptarse a las preferencias y hábitos del usuario y puede ser compartido por múltiples usuarios.

- Poseen mayor robustez y mejor tolerancia a los fallos (un agente puede fallar en el sistema sin resultados catastróficos). También proporciona una mayor flexibilidad y adaptabilidad y gracias a ellos se obtienen tiempos de respuesta menores que los obtenidos con los sistemas clásicos IA (al poder reaccionar directamente ante los sensores-estímulos sin necesidad de hacer un procesamiento y planificación previa en un modelo de representación interno).

- Asignan dinámicamente las tareas mediante la solicitud, introducción y eliminación dinámica de agentes en el sistema y a través de un mecanismo fiable para realizar un control distribuido.

- Permiten que dos agentes se comuniquen en la misma localización y no a través de la red (reduciendo de esta forma las transferencias de datos por la red).

- Reducen la complejidad del desarrollo del agente debido a que la movilidad la proporciona la plataforma.

- No necesitan efectuar accesos y procesamiento de estructuras de representación, por lo que son más rápidos que los sistemas que sí tienen que hacerlo y pueden también utilizarse para cumplir restricciones de tiempo real estricto.

- Facilitan la reparación del tiempo necesario para las comunicaciones del tiempo de procesamiento, pudiéndose realizar este último bajo restricciones de tiempo real estricto a través del nivel reactivo sin las interrupciones debidas a los procesos de comunicación utilizados en los métodos tradicionales.

Desventajas de los Agentes de Software Móviles

- Su principal inconveniente reside en que algunas técnicas de gestión de agentes no detectan ni resuelven los conflictos que pueden darse entre las tareas. Se facilita así la existencia de tareas y peticiones contradictorias, lo cual no puede permitirse en coordinación.

- Las técnicas de multiagente requieren que los agentes sean capaces de manejar una gran cantidad de información. De hecho, son necesarias más comunicaciones y una mayor potencia de procesamiento que en el resto de técnicas.

- Puede surgir un cuello de botella importante si el número de agentes es elevado, incluso si el sistema de gestión está dividido.

- La necesidad de que los agentes hagan un uso racional de las estrategias definidas para maximizar las utilidades.

- El hecho de que en la negociación los agentes sólo consideran el estado actual y no las acciones pasadas ni futuras.

- La suposición de que todos los agentes tienen el mismo modelo interno y las mismas características.

Conclusiones

El gran potencial de los agentes desarrollados es el poder que tienen de interactuar entre ellos sobre plataformas, protocolos de transferencia, lenguajes de programación y lenguajes de comunicación diferentes.

Los sistemas de agentes de software móvil proporcionan una plataforma que ofrece un mayor nivel de abstracción con respecto a la arquitectura distribuida que conocemos. De esta forma nos podemos acercar aun más a los requisitos del usuario, ya que el programador puede expresar las necesidades de una forma más flexible.

La colaboración entre agentes se realiza mediante un lenguaje de comunicación (sintaxis, semántica y pragmática) que soporta la capacidad de compartir conocimiento.

Los agentes móviles son programas escritos normalmente en lenguajes interpretados como Java que pueden ser enviados desde una maquina cliente a una maquina servidor. Esta comunicación se puede lograr mediante los procesos de migración o clonación.

Los costos en el desarrollo e implantación de la tecnología de agentes móviles varían de acuerdo a los requerimientos y necesidades de las empresas. La infraestructura en el área de las comunicaciones es un factor muy importante a tener en cuenta.

En el futuro se llegará un estado en el que un agente podrá trasladarse sin dificultad de una maquina a otra dentro de un sistema totalmente heterogéneo. Java es uno de los lenguajes de programación que aporta muchas de la características requeridas por la tecnología de agentes móviles tanto en comunicaciones como en seguridad.

Se puede concluir que el objetivo ideal de la tecnología de agentes es modelar una entidad *software* con las características propias de una persona, de manera que la resolución de un problema sea lo más parecida posible a su resolución en el mundo real.

Glosario

Acción (*action*): Elemento básico que representa una actividad que un agente puede realizar. Una clase especial de acción es un acto de comunicación

Agente ARB (*ARB Agent*): Agente que proporciona el servicio ARB (*Agent Resource Broker*). Este servicio permite a un agente usar otros servicios fuera del mundo de los agentes (sistemas propietarios o "*legacy systems*").

Agent Communication Language (ACL):

Lenguaje con una sintaxis, semántica y pragmática formalmente definidas. Es la base de las comunicaciones entre agentes de una naturaleza heterogénea.

Agent Communication Channel (ACC) Router: Es un agente que usa la información proporcionada por el AMS para encaminar los mensajes entre agentes dentro de la misma plataforma o entre plataformas diferentes.

Agent Management System (AMS): Es un agente que gestiona la creación, destrucción, suspensión, autentificación y migración de los agentes de la plataforma. Además proporciona un servicio de nombres para todos los agentes que residen en ella. Guarda la correspondencia entre GUID (*Globally Unique Identifier*) del agente y la dirección de transporte donde se encuentra.

Agent Platform (AP): Proporciona la infraestructura necesaria para que los agentes puedan ser utilizados. Un agente debe ser registrado en una plataforma para poder interactuar con otros agentes de esa plataforma. Un AP contiene tres componentes básicos: ACC, AMS y DF.

ARA: Agentes para la acción remota. Agentes especializados en viajar remotamente a través de la red.

CORBA: *Common Object Request Broker Architecture*. Un estándar bien establecido que permite la comunicación de sistemas de objetos distribuidos.

COOL: Es un lenguaje para la coordinación que modeliza la conversación entre los agentes mediante una máquina de estados finitos. Cada estado representa el punto de la conversación en la que nos encontramos.

Directory Facilitator (DF): Es un agente que proporciona un servicio de páginas amarillas. Guarda información de los agentes y de los servicios que éstos ofrecen.

FIPA: *Foundation for Intelligent Physical Agents*. FIPA se ha convertido en el estándar con más repercusión y aceptación social. Trata todos los temas relacionados con los sistemas multiagente proporcionando una amplia gama de documentos.

FIPA ACL: Toda la plataforma FIPA se basa en la comunicación entre agentes mediante ACL (*Agent Communication Language*). Un mensaje ACL consiste básicamente en una expresión de un determinado lenguaje con un conjunto de términos específicos propios de la ontología usada.

GIOP: *General Inter-ORB Protocol*. Es el protocolo de comunicación estándar entre ORB's establecido por la especificación CORBA.

GUID: *Globally Unique Identifier*. Es la identidad propia del agente, es decir, es el nombre del agente.

HAP: *Home Agent Platform*. Es la plataforma en la cual se creó un agente. La responsabilidad del HAP es garantizar la identidad del agente en sus relaciones con otros agentes y plataformas.

IDL: *Interface Definition Language*. Lenguaje estandarizado por CORBA. OMG IDL es un lenguaje declarativo estándar. El léxico del lenguaje OMG IDL obedece a las mismas reglas que el léxico del lenguaje C++.

IPMT: *Internal Platform Message Transport*. Método de intercambio de mensajes dentro de la misma plataforma. Depende de la implementación.

JATLITE: *Java Agent Template*. Es un témplate de la universidad de Stanford. Estándar de agentes desarrollado en Java.

KIF: *Knowledge Interchange Format*. Es un lenguaje de contenido y equivale a la sintaxis en el lenguaje de comunicación común.

KQML: *Knowledge Querying and Manipulation Language*. Equivale a la pragmática en un lenguaje de comunicación común.

KSE: *Knowledge Sharing Effort*. Es una organización de estandarización que busca compartir el conocimiento entre agentes.

MAFAgentSystem: El interfaz *MAFAgentSystem* define las operaciones de control de los agentes.

MAFFinder: El MAFFinder es un objeto servidor de nombres. Antes de que un cliente pueda solicitar al objeto *MAFFinder* la búsqueda de un objeto, el cliente tiene que obtener el objeto referencia del mismo *MAFFinder*.

MASIF: *Mobile Agent System Interoperability Facilities*. Corresponde a la OMG. Es una organización de estandarización internacional para agentes. Considera a los agentes como objetos CORBA que tienen la posibilidad de moverse, y ejecutarse autónoma y asíncronamente en sistemas de ejecución seguros (sistemas de agentes).

MOA: *Mobile Object Agent*. Es una plataforma que soporta agentes de software móviles.

OMG: *Object Management Group*. Organización Internacional que desarrolló la especificación CORBA.

Ontología (Ontology): Una ontología da significado a los símbolos de un determinado dominio de conversación. Para que dos agentes puedan entenderse es necesario que ambos esten dando el mismo significado a los símbolos de los mensajes intercambiados.

ORB: *Object Request Broker*. Permite que el agente use otros servicios fuera del mundo del objeto.

Protocolo (Protocol): Un patrón de mensajes intercambiados entre agentes para realizar una determinada tarea. Son normalmente usados para simplificar la lógica que es necesario implementar para posibilitar el diálogo entre agentes.

Proxy: Objeto que proporciona una interfaz local a un objeto remoto. Redirige las peticiones al objeto remoto.

UML: Lenguaje de Modelado Unificado. Lenguaje gráfico estándar para visualizar, especificar, construir y documentar el modelado de un sistemas de software.

XDR: *eXternal Data Representation* de SUN. Representación externa estandarizada para transmitir y recibir valores.

Referencias Bibliográficas

1. JENNINGS, N. R., SYCARA, K., WOOLDRIDGE, M. "A Roadmap of Agent Research and Development", *Autonomous Agents and Multi-Agent Systems*, I, 1998.
2. *Diccionario de la Lengua Española*, Real Academia Española, XXI edición, 1992.
3. *MASIF-RTF Results*, Object Management Group, 1998.
4. RUSSEL, S., NORVING, P., *Artificial intelligence: A Modern Approach*, Prentice Hall, 1995.
5. FRANKLIN, S., GRASSER, A., *Is it an Agent or just a Program? A Taxonomy for Autonomous Agents*, Universidad de Memphis, 1996. http://www.msci.memphis.edu/~franklin/AgentProg. html.
6. FIPA Specification http://www.fipa.org
7. Object Management Group http://www.omg.org
8. *Specification of KQML Agent-Communication Language plus example agent policies and architectures*, The DARPA Knowledge Sharing Initiative, External Interfaces Working Group, 1993. http://www.cs.umbc.edu
9. *FIPA Abstract Architecture Specification* and *Agent Management Specification*, http://www.fipa.org.
10. Agent Society Home Page http://www.agent.org
11. GUERRAOUI, Ravhid, FAYAD, Mohamed E., *Oriented Object Distributed Programming Is Not Distributed Oriented Object Programming* 1999.
12. VENNERS, B., *Solve Real Problems with Aglets, a Type of Mobile Agents*, JavaWorld, mayo 1997. http://www.javaworld.com/javaworld/jw-05-1997/jw-05-hood.html
13. CARAMAZANA CÁRCAMO, Alberto, Agentes Moviles. Consultor Senior de Ideal Objects.
14. FERNÁNDEZ MOYA, Francisco, fmoya@inf-cr.uclm.es, *Introducción a la arquitectura corba para sistemas distribuídos*, Grupo de Arquitectura y Redes de Computadores, 2001.
15. DIOSA, Henry Alberto, *Propuesta para la conformación de grupo de trabajo en procesamiento basado en objetos distribuidos con CORBA y Objetos de Software Móviles Usando Java*, Universidad Distrital Francisco José de Caldas, Bogotá, agosto de 2000.
17. POSADA YAGÜE, Juan Luis, *Arquitectura de Procesos en sistemas Reactivos Distribuidos*, Departamento de informática de sistemas y computadores, septiembre de 2000.
18. GÓMEZ LABRADOR, Ramón M., *Agentes Móviles y Corba*, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, febrero de 1999. (ramon.gomez@fie.us.es)
19. TANENBAUM, Andrew S., *Redes de Computadoras*, Prentice may, Tercera Edición, México, 1997.

Comunications Magazine, IEEE, July 1998.

Mobile Software Agents for Telecomunications.

Htpp://www.comsoc.org

Asociados de comunicaciones.

Htpp://www.research.digital.com/SRC/personal/Luca Cardelli/Obliq

Htpp://www.cs.uit,no/DOS/Tcoma

Htpp://www.meita.com/HSL/Projects/Concordia

Htpp://www.tinac.com

Htpp://www.agent.org

Organización de Agentes Móviles

Htpp://www.cs.dartmouth.edu

Htpp://www.omg.org/library/schedule/technology