**INTRODUCCIÓN AL ESTÁNDAR FIPA**

Se pretende introducir el estándar FIPA (Foundation for Intelligent Physical Agents).

Es un estándar ampliamente reconocido y se ha convertido en un punto de referencia inevitable.

FIPA es ahora mismo uno de los Comités de estándares de IEEE.

http://www.fipa.org

<http://www.fipa.org/fipa99>

**Breve repaso histórico**

FIPA comenzó sus actividades en 1995, con el convencimiento de que algunos aspectos de la tecnología de agentes ya estaban suficientemente maduros como para considerar su estandarización.

El proceso de estandarización empezó creando un consorcio internacional (universidades e industrias) que desarrollaran un estándar de carácter público.

Este estándar se ocuparía de describir el comportamiento externo y las interfaces de los agentes y sistemas de agentes.

Un primer borrador del estándar (FIPA 97) se publicó en Octubre de 1997 en Munich.

**Organización del estándar**

El estándar FIPA se compone de 2 partes claramente diferenciadas: **Referencias Tecnológicas** y **Referencias a Aplicaciones**.

 La primera parte (*normativa*) se ocupa de especificar una plataforma tecnológica que permita el desarrollo de sistemas multi-agente abiertos.

La segunda parte (*informativa*) presenta algunas aplicaciones reales que motivan el uso de estos sistemas.

Aquí nos centraremos principalmente en el área tecnológica.

**Referencias Tecnológicas (Normative Specification)**

Esta parte del estándar se ocupa de presentar las normas que deben regir el comportamiento e interfaces externas de un sistema de agentes y asegura la interoperabilidad con otros sistemas que cumplen dichas normas.

El estándar no impone (aunque en algunas ocasiones da recomendaciones) como se deben implementar estos sistemas, sino que asegura que independientemente de las opciones tomadas a la hora de ser implementados van a ofrecer un comportamiento similar asegurando una total interoperabilidad.

En su versión incial (FIPA 97) tres documentos integraban esta parte del estándar:

Part 1: *Agent Management*.

Part 2: *Agent Communication Language.*

Parte 3: *Agent/Software Integration.*

Posteriormente se han ido actualizando e incluyendo nuevos documentos hasta alcanzar la versión con la que actualmente se trabaja:

Part 0: *Architectural Overview.*

Part 8: *Human/Agent Interaction.*

Part 10: *Agent Security*.

Part 11: *Agent Mobility.*

Part 12: *Ontology Services.*

Part 13: *FIPA Developer’s Guide*.

Part 14: *Nomadic Application Support*.

Part 16: *Agent Message Transport.*

Part 17: *Agent Naming.*

Part 18: *FIPA Content Languages.*

**Referencias a Aplicaciones (Informative Specification)**

Esta parte del estándar muestra cómo se puede utilizar la tecnología del estándar en aplicaciones industriales.

Pretende dar unas pistas de por donde los sistemas multi-agente pueden introducirse a nivel comercial, importante si se quiere que estos sistemas realmente tengan futuro.

Las partes que se incluyen son:

Part 4: *Personal Travel Assitance.*Define un sistema de agentes que asiste al usuario a planificar un viaje, incluyendo elección de itinerarios y medios de transporte, reserva de billetes y hoteles, y ayuda durante el viaje dando información actualizada en cada momento.

Part 5: *Personal Assistant.*Facilita las labores del usuario que se realizan de un modo regular y repetitivo durante su trabajo diario en la oficina, incluyendo la gestión del correo electrónico diario, las labores de administración del sistema, etc*.*

Part 6: *Audio/Video Entertainment & Broadcasting.*Ayudan al usuario de sistemas digitales de entretenimiento (tanto bajo broadcast como  bajo demanda) a seleccionar entre los cientos e incluso miles de posibilidades que se le ofrecen.

Los agentes de usuario colaborarían y negociarían con los agentes del proveedor del servicio para adecuar la programación a los intereses del cliente de acuerdo al perfil e intereses del mismo.

Part 7: *Network Management & Provisioning.*Numerosos proveedores de servicios de telecomunicaciones, ofrecen servicios que se componen de elementos pertenecientes a diferentes operadores.

El objetivo es ofrecer en cada momento las mejores condiciones del servicio en cuanto a calidad y coste.

El sistema permite una gestión descentralizada aplicado principalmente a las Redes Privadas Virtuales (VPNs).

Part 9: *Product Design & Manufacturing.*Presenta una aplicación que recoge el proceso de fabricación de un producto en todas sus fases. Pretende utilizar la mayor parte del estándar: Ontologías, Interacción con el usuario, Seguridad y Movilidad.

**Gestión de Agentes**

**Introducción**

A la hora de definir la plataforma de agentes (AP-*Agent Platform*), FIPA

1. define solo el comportamiento externo (interfaz), dejando las decisiones de diseño a cada equipo de desarrollo.
2. establece un sistema totalmente abierto de tal forma que sistemas heterogéneos puedan interactuar a nivel de sociedades de agentes.

La AP proporciona un marco en el que los Agentes FIPA puedan vivir en sociedad.

Establece el modelo lógico referente a la creación, destrucción, registro, localización y comunicación de agentes.

El gráfico siguiente representa el modelo de referencia que la FIPA ha diseñado para su sistema de agentes.



**Agent Platform (AP)**

Es el núcleo de modelo de referencia de FIPA, y proporciona la infraestructura en la cual los agentes pueden ser desarrollados y usados.

Está constituida por todo el hardware y software (S.O., software de comunicaciones, middleware y software de agente) necesario para llevar a cabo esta infraestructura.

La idea clave, es que cada implementación tenga libertad en cuanto las componentes usadas, pero manteniendo siempre una imagen estandarizada hacia el exterior.



También hay que reseñar que todos los servicios que define FIPA, excepto el de transporte de mensajes (IPMT), van a ser agentes preparados para proporcionar dichos servicios.

Esto supone que la comunicación con ellos va a ser mediante mensajes ACL (*Agent Communication Language* ) usando la ontología definida para ese servicio.

**Agent Management System (AMS)**

El AMS es un componente que debe aparecer en toda plataforma de agentes FIPA.

Su cometido esconocer en todo momento el estado de su plataforma y de los agentes que pertenecen a ella.

Servicios que ofrece:

* creación, destrucción y control del estado de los agentes,
* supervisión de los permisos para que nuevos agentes se registren en la plataforma,
* control de la movilidad de los agentes,
* gestión de los recursos compartidos y
* gestión del canal de comunicación.
* de nombres (*Agent Name Service-ANS)(* servicio de *páginas blancas)*

Este servicio de nombres asocia el nombre que identifica a un agente en su sociedad y la dirección de transporte real en la que se encuentra.

Proporciona un método básico para buscar un agente dentro de la plataforma.

Como elemento de gestión, el AMS permite controlar el ciclo de vida de los agentes.



**Directory Facilitator (DF)**

Como complemento al Servicio de Nombres, AP debe incluir un servicio de *páginas amarillas*, que permita buscar un agente por sus capacidades y no por su nombre.

Esta labor la realiza el *Directory Facilitator* (DF).

Los agentes se registran en él indicando los servicios que ofrecen.

Cuando otro agente tiene unas necesidades concretas lanza una búsqueda del servicio deseado obteniendo los agentes que le ofrecen estos servicios.

**Agent Communication Channel (ACC)**

Todos los agentes FIPA deben tener acceso a un ACC.

Está encargado de gestionar el envío de mensajes entre agentes (de una misma o de distintas plataformas).

Esta gestión consiste principalmente en el encaminamiento de los mensajes ACL desde el agente origen al agente destino.

Cuando un agente quiere enviar un mensaje a otro en una localización remota, entrega este mensaje al ACC que se encargará de entregárselo al ACC de la plataforma donde se encuentre el otro agente.

Este último ACC notifica al agente destino que ha recibido un mensaje para él.

El modelo de comunicación entre agentes es asíncrono.

Esto implica que:

* El ACC no se queda bloqueado ante el envío o recepción de mensajes.
* Existen colas de envío y recepción de mensajes.
* Políticas de gestión de colas de mensajes.

**4.6.- Internal and Agent Platform Message Transport (IPMT-APMT)**El *Internal Platform Message Transport* (IPMT) representa toda la infraestructura de comunicaciones que va a permitir que dos agentes dentro de la misma plataforma puedan comunicarse. El *Agent Platform Message Transport* (APMT) es otro elemento que hace la misma labor que el IPMT pero a nivel de comunicaciones entre plataformas (ACCs). En principio FIPA no decía nada sobre cómo implementar estos dos elementos de bajo nivel por ser muy dependientes del entorno hardware/software donde se estaba trabajando. Pero con la parte 16 del estándar que recoge el *Agent Message Transport* (AMT) quiere unificar ambos elementos de comunicación en uno único a la vez que da unas líneas básicas de cómo debe ser la arquitectura del mismo.

**5.- Agent Communication Language**

**5.1.- Introducción**En una sociedad multiagente, los agentes deben cooperar para realizar sus tareas y alcanzar sus objetivos. Los mecanismos de comunicación tradicionales permiten que dos agentes puedan transferirse información, pero eso es insuficiente cuando el objetivo es el conseguir un comportamiento social. Es decir, necesitamos otros mecanismos que nos permitan dotar a los mensajes intercambiados de un contenido semántico. Hasta ahora los métodos de intercambio de comunicación utilizados, como son los *sockets*, se basan en el envío de mensajes con información en algún formato determinado que los extremos de dicha comunicación son capaces de procesar. Cada aplicación tiene que tener un procedimiento que permite procesar la información y realizar las tareas asociadas a la misma. Se está realizado una solución *ad hoc* que impide incluir dicho procesamiento en el propio mecanismo de comunicación. Con FIPA el canal de comunicación es algo más complejo pues además de la información binaria se está enviado conocimiento por el hecho de que esta información contiene mensajes ACL que son comprendidos por cualquier agente del sistema. Para definir el modelo de comunicación, FIPA parte de la idea de que los agentes poseen la capacidad de razonamiento. Esta capacidad viene dada por una serie de actitudes mentales que se definen como:

**Creencias** : Conjunto de proposiciones que el agente acepta como verdaderas. Es decir, lo que el agente sabe del “mundo”.

**Objetivos**: Denota una propiedad o conjunto de propiedades del mundo que el agente quiere que sean verdaderas, pero que actualmente no están entre sus creencias.

**Intenciones**: Conjunto de acciones planificadas por el agente que le permitan llegar a un estado deseado. Mediante sus acciones, un agente puede contribuir a cambiar el estado de su entorno y así satisfacer sus objetivos.

Un agente influye en el conocimiento y acciones de otros agentes mediante un “acto de comunicación” (*speech act*), que se realiza mediante el envío de un mensaje desde el origen al destino.

Ambos extremos de la comunicación deben entender el mensaje, y éste debe ser algo más que un intercambio de datos, debe tener asociado un contenido semántico accesible por ambas partes.

FIPA ha desarrollado el lenguaje ACL (*Agent Communication Language*), una evolución de lenguaje KQML (*Knowledge Query Manipulation Language*), tomado como estándar de facto durante mucho tiempo para el intercambio de conocimiento entre agentes.

La definición semántica formal de los mensajes ACL ha sido uno de los esfuerzos mayores dentro del estándar FIPA, y otorga a este lenguaje de una gran aceptación como estándar dentro del mundo de los agentes.

A partir de ahora hay que entender un agente como una entidad que entiende el lenguaje ACL y es capaz de intercambiar conocimiento con otros agentes mediante el uso del mismo.

Ya no se habla de que la interfaz de un agente (objeto) ofrezca una serie de servicios que se pueden invocar o instanciar, sino que un agente va a proporcionar ciertos servicios que se le pueden solicitar mediante un mensaje ACL que incluya dicha solicitud.

Como se puede ver se pasa de un modelo de objetos a un modelo de agentes inteligentes basados en actos de comunicación.

**5.2.- Estructura de un mensaje ACL**

Un mensaje ACL está formado por:

**Identificador del tipo de comunicación**. Define el significado principal de mensaje. Es decir, lo que el agente origen pretende con dicho mensaje.

Ejem: *inform*, *request*, *agree*, *query*.

**Secuencia de parámetros del mensaje**. Es un conjunto de parejas clave-valor que permiten asociar a cada acto de comunicación concreto toda la información necesaria. A continuación se muestra un ejemplo de mensaje ACL:

**( inform :sender** *agent1***:receiver** *agent2***:content***(price book01 1000)***:in-reply-to** *round-4***:reply-with** *bid04***:language** *sl***:ontology** *myontology***)**

Este mensaje enviado del agente *agent1* al agente *agent2* informa que el precio del libro *book1* es de 1000. utilizan el lenguaje *sl* y la ontología *myontology* para describir esta información.

**5.2.1.- Tipos de mensaje**Los tipos de mensajes del lenguaje ACL definidos por la FIPA son los siguientes:

*accept-proposal*

*agree*

*cancel*

*cfp*

*confirm*

*disconfirm*

*failure*

*inform*

*inform-if*

*inform-ref*

*not-understood*

*propose*

*query-if*

*query-ref*

*refuse*

*reject-proposal*

*request*

*request-when*

*request-whenever*

*subscribe*

A modo de ejemplo se pueden describir algunos de estos tipos de mensajes:

*request:* Un agente solicita a otro que realice una acción. *agree:* Un agente acepta la petición realizada.

*refuse:* Un agente rechaza la petición realizada.

*failure:* Un agente informa de que ha habido un problema al realizar la acción.

Para ver el modelo semántico de cada tipo de mensaje ACL ver “*FIPA Specification, Part 2 - Agent Communication Language”.*

**5.2.2.- Parámetros del mensaje**Los parámetros básicos del lenguaje ACL definidos por la FIPA son los siguientes:

**:sender**Indica el origen del mensaje ACL. Tiene asociado la dirección de un agente.

**:receiver**Indica el destino del mensaje ACL. Tiene asociado la dirección de un agente.

**:content**Tiene asociado el contenido del mensaje. Debe ser una expresión del lenguaje definido por el parámetro *:language* usando símbolos de la ontología indicada por el parámetro *:ontology*.

**:reply\_with**Indica que valor debe asociarse al campo *:in\_reply\_to* en el mensaje de respuesta.

**:in\_reply\_to**Permite seguir el thread del diálogo junto con el campo *:reply\_with.*

**:envelope**No usado. Se reserva para llevar información de transporte (tiempos, rutas, QoS,...)

**:language**Lenguaje de contenidos con el que se ha formado la expresión en *:content*. El estándar en la versión 99 define el lenguaje de contenidos de carácter general SL y tres subconjuntos del mismo: SL0, SL1, y SL2.

**:ontology**Ontología que define los símbolos con los que se ha formado la expresión en *:content*.

**:reply-by**Tiempo máximo de espera del mensaje de respuesta.

**:protocol**Protocolo o patrón de comunicación que está siendo usado.

**:conversation-id**Sirve para identificar la secuencia de mensajes enviados en una conversación.

**5.3.- Protocolos de comunicación entre agentes**

Las conversaciones entre agentes suelen seguir ciertos patrones en la secuencia de mensajes intercambiados.

Cada una de estas secuencia típica se denomina **protocolo de comunicación**.

Como se ha visto, el campo *:protocol* de un mensaje identifica el protocolo de comunicación que se está utilizando por ambas partes en un determinado momento de la conversación.

Los protocolos estándar definidos por FIPA son los siguientes:

*FIPA-request*

*FIPA-query*

*FIPA-request-when*

*FIPA-contarct-net*

*FIPA-interated-contract-net*

*FIPA-auction-english*

*FIPA-auction-ducth*

El siguiente diagrama muestra como ejemplo el esquema del protocolo *FIPA-request*:

Este protocolo permite a un agente solicitar que otro realice una acción, y el receptor informa de que ha realizado la acción con un mensaje de tipo *inform*, o en el caso de que no haya podido indica la causa con un mensaje de tipo *failure*.

**6.- Agent/Software Integration**

En el futuro los sistemas basados en agentes tendrán que convivir e interoperar con sistemas tradicionales.

Hay gran cantidad de aplicaciones como sistemas de tarificación, bases de datos de usuario, controladores de dispositivo o herramientas de escritorio. que van a perdurar por mucho tiempo y que por tanto deben ser integrados en posibles sistemas multi-agente.

Hasta ahora se habían llevado soluciones *ad hoc* para lograr dicha integración, de tal forma que en cada ocasión había que inventar tal solución. FIPA ha reconocido este problema y ha propuesto un mecanismo estándar para dotar de cierta universalidad y reusabilidad a los sistemas tradicionales dentro de las sociedades de agentes.

Se definen dos agentes para la gestión del software o servicios no basados en agentes:

***Wrapper agent***Encapsula este software y lo eleva al nivel de la sociedad de agentes.

***Request Broker***Realiza la labor de directorio o páginas amarillas para los servicios encapsulados.

El modelo usado es el de instanciación de servicios.

Un agente que desea hacer uso de parte de software encapsulado puede solicitar al *request broker* información de los servicios disponibles a través del *wrapper agent,* y una vez conocido puede instanciar uno de tales servicios.

Toda la negociación e intercambio de información se va a hacer mediante el lenguaje ACL puesto que estamos en el nivel del sistema de agentes, mientras que el software tradicional va a seguir ejecutándose de forma habitual en un nivel inferior.

El estándar no acomete el problema de la interconexión entre el *wrapper agent* y el software tradicional.

Esto se delega a cada diseño o implementación concreta del estándar por no afectar al comportamiento exterior del sistema. Y por supuesto este problema no es trivial ni mucho menos.

**7.- Human/Agent Interaction**

La interacción entre los usuarios de un sistema de agentes y los propios agentes es un problema bastante complejo y difícil de estandarizar. Por eso en la versión 97 del estándar no se acometió esta labor.

Según ha ido madurando el estándar se ha visto la necesidad de que una parte del mismo debía reflejar al menos de forma conceptual el cómo abordar el problema.

La siguiente figura ilustra las diferentes entidades (y sus relaciones) que son consideradas como cruciales para los procesos de interacción entre usuarios y agentes.

La parte superior representa el mundo real del usuario mientras que la parte inferior representa el mundo donde los agentes operan y se comunican.

Agent 1 DF Agent 2 User B User A UDMA UPA **FIPA-HAI Scope**GUI Natural Language Speech etc **ACC fipa-udms fipa-ups fipa-udms fipa-ups**ACL Interface non-ACL Interface FIPA user dialogue management service ontology FIPA user personalization service ontology

Dos son los servicios (agentes) que permiten llevar a cabo estos procesos de integración entre el mundo real y la sociedad de agentes:

**UPS** (*User Personalization Service*) Parte del comportamiento “inteligente” de un agente se basa en el modelado de sus usuarios. Este modelado guarda información adquirida a través de su interacción: preferencias, deseos, conocimientos y objetivos.

El **UPA** (*User Personalization Agent*) pemite registrar, actualizar, consultar y gestionar todo lo referente a los modelos de los usuarios que interaccionan con el sistema.

**UDMS** (*User Dialog Management Service*) Recubre los diferentes tipos de componentes software de interacción con el usuario (GUI, reconocedores de lenguaje natural, reconocedores del habla,etc.). Es un *gateway* entre el mundo de los humanos y el mundo de los agentes (ACL).

El **UDMA** (*User Dialog Management Service*) interacciona con el UPA mediante ACL para mantener el modelo de los usuarios que interaccionan con el sistema

**8.- Agent Security**

**8.1.- Introducción**

En cualquier sociedad que se precie debe haber un conjunto de mecanismos de seguridad que asegure el buen comportamiento de todos los miembros de esa sociedad. Esto también debe ser cierto cuando la sociedad es de agentes software.

La dificultad para resolver todos los problemas de seguridad que aparecen en un sistema tan complejo como es un sistema multi-agente, es uno de los motivos por los que esta tecnología no se está usando en la industria para la resolución de problemas reales.

FIPA ha detectado la mayoría de estos problemas pero el resolverlos no es tan simple como sería deseable.

El problema no aparece solo en este tipo de entornos. Hoy en día, con el gran desarrollo de Internet, están apareciendo fuertes requerimientos en cuanto a la seguridad de los sistemas, y en muchos casos las soluciones todavía están por llegar.

Los problemas de seguridad son de una complejidad excepcional cuando hablamos de sistemas de agentes móviles, en los que éstos pueden migrar ejecutándose en diferentes nodos de la red.

Como en otras partes del estándar, FIPA no gasta sus esfuerzos en buscar soluciones nuevas para los problemas, sino en como encajar soluciones existentes y de gran aceptación, dentro de su plataforma de agentes.

La recomendación en este caso es el estándar de seguridad X.509 basado en infraestructuras de seguridad de clave pública (PKI).

El estándar X.509 presenta un modelo de seguridad asimétrico o basado en certificados de clave pública y clave privada. Algunos conceptos relevantes de este estándar son los siguientes:

**Autentificación**: Capacidad de verificar que la entidad que ofrece sus credenciales (certificado) es quien dice que es.

**Certificado**: Información que prueba que una identidad es quien dice que es. En el X.509 se permiten distintos tipos de certificados basados en login y password, huellas dactilares digitalizadas, imagen escaneada de la retina, propiedades del habla,etc.

**Autoridad certificadora**: Una entidad que genera y determina cuando un certificado es válido. Todas las demás entidades deben poder confiar en la veracidad de la información que proporciona esta autoridad.

**Modelo de certificación**: El mecanismo usado para autentificar las credenciales.

**Firma digital**: Los datos son firmados con un certificado para que el destino pueda identificar el origen del mismo y asegurar la integridad (nadie lo ha modificado) del mismo.

**Encriptar**: Habilidad de transformar la información a un formato ilegible de tal forma que sólo pueda ser recuperado por el poseedor de un determinado certificado.

**Identidad**: Entidad que puede ser unívocamente identificada mediante un certificado. Dicha entidad debe estar dada de alta al menos en una autoridad certificadora reconocida.

El modelo de certificación que presenta el estándar X.509 se basa en el uso de 2 tipos de claves, privadas a las que solo tiene acceso la entidad propietaria del certificado, y públicas a las que tienen acceso todas las demás entidades de la sociedad.

Por cada entidad, la autoridad certificadora correspondiente genera un par de claves (pública y privada) para el cifrado (confidencialidad) y otro par de claves (pública y privada) para la firma (autenticidad).

Cada entidad recibe sus claves privadas por parte de la autoridad certificadora. Las claves públicas de todas las entidades están guardadas en alguna estructura de directorio (LDAP) y pueden ser consultadas en cualquier momento.

También es posible que una entidad distribuya sus claves públicas a las entidades con las que quiere comunicarse de forma segura.

Cuando el emisor E envía un mensaje al receptor R:

Realiza un **resumen del mensaje** (hash) de tamaño prefijado.

Lo **cifra** con la **clave pública** de la entidad R. De este modo el contenido del mensaje solo podrá ser decodificado por R cuando **descifre** el mensaje con su **clave privada** de cifrado (que solo R conoce).

Lo **firma** con su propia **clave privada** de firmado, para que cuando R reciba el mensaje pueda comprobar que realmente ha sido enviado por E. para ello usa la **clave pública** de firmado de E que debe de conocer.

Este modelo de seguridad se presenta a nivel de agentes. Son posibles otros modelos de seguridad de más bajo nivel como son SSL (*Secure Socket Layer*) a nivel de la capa de transporte.

**Problemas detectados**

Algunos de los problemas detectados por la FIPA y que deben ser resueltos en futuras actualizaciones del estándar se enumeran a continuación:

**Copia de Código y datos**. Los sistemas de agentes se suelen implementar usando lenguajes interpretados como Java o TCL para conseguir una mayor portabilidad e interoperabilidad. Pero esto supone que el código y los datos usados son fácilmente accesibles y reproducibles.

**Alteración de código y datos**. De forma similar a la copia, la alteración de código y datos de un agente puede ser un problema potencial. Una plataforma de agentes puede estar preparada para atacar a los agentes que lleguen a ella.

**Ataques en grupo**. Aunque una plataforma este protegida ante el ataque de algún agente con malas intenciones, son posibles otros tipos de ataque en grupo complicados de evitar.

**Copia y reenvío**. Un mensaje puede ser copiado y enviado a otros agentes no incluidos en la lista de destinos del mismo.

**Denegación del servicio**. Una plataforma de agentes puede negar ciertos servicios a un agente sin motivo aparente.

**Rechazo de responsabilidad**. Es muy complicado repartir responsabilidades cuando ocurre un problema. Una plataforma o agente puede actuar de forma maliciosa y no admitir su responsabilidad.

**Mascarada**. Un agente puede hacerse pasar por otro en una conversación para obtener información o para acceder a servicios que de otra forma no tendría accesibles.

**Liquidez**. En algunos entornos se han creado sistemas de pago (tickets) para que un agente tenga que pagar para conseguir un servicio. Esto implica que se debe gestionar la liquidez de los agentes y que el proveedor del servicio ofrezca tal servicio una vez cobrado.

**9.- Agent Mobility**

**9.1.- Introducción**Los sistemas móviles (*Wireless Systems*) están aumentando de manera vertiginosa debido a la necesidad, cada vez mayor, que tienen los usuarios de acceder a las vías de información en cualquier lugar donde se encuentren y el tener un canal de comunicación siempre disponible.

Esto está produciendo una explosión en la venta de telefonía móvil, PDAs, ordenadores portátiles, etc.

Estos sistemas no tienen una conexión de red permanente y cuando la tienen ésta puede ser de muy baja calidad.

Los agentes móviles permiten resolver los problemas que originan este tipo de sistemas.

Un agente móvil añade a las características de un agente la capacidad de moverse desde una máquina a otra en redes heterogéneas bajo su propio control.

Puede suspender su ejecución en cualquier momento, transportarse a otra máquina y restablecer la ejecución en su nueva localización como si nada hubiese pasado.

Cuando el agente ha realizado su tarea vuelve a su lugar de origen para devolver los resultados, aprovechando algún momento en el que la conexión de red esté establecida.

También se están usando en entornos donde el control distribuido es un requisito o donde el consumo de ancho de banda se reduce enormemente enviando un agente a la otra máquina en vez de comunicarse por la red.

La tecnología de agentes móviles se puede considerar un campo de investigación que ha seguido un desarrollo paralelo al de los agentes inteligentes.

Hasta ahora se estaban haciendo esfuerzos por separado pues ambas tecnologías estaban poco maduras.

Ahora parece que es posible integrar ambas partes para conseguir un modelo de agentes que aporte todas las capacidades deseables.

**9.2.- Movilidad en la plataforma FIPA**

Siguiendo la filosofía del estándar, FIPA no propone soluciones tecnológicas sino más bien aportar ideas en la forma de integrar de forma homogénea y natural la capacidad de los agentes móviles dentro de su plataforma.

 La interoperabilidad entre sistemas es muy importante en este campo debido a que si dos sistemas de distinto origen no pudiesen intercambiar agentes la movilidad se quedaría como una capacidad no aprovechable realmente.

Cuando se trata de agentes móviles realmente podemos trabajar con dos soluciones más o menos complejas de llevar a cabo:

**Migración**: Este modelo es más complejo ya que requiere la transferencia de código y datos del agente a la plataforma remota. Cuando un agente va a migrar se suspende, se transfiere por la red y se despierta en el otro extremo siguiendo su ejecución en el mismo punto donde lo había dejado.

**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 ordenes que el clon debe realizar.

Como para otros servicios prestados en la plataforma de agentes, el estándar define un conjunto de mensajes ACL junto con la Ontología usada para los mismos (ver *FIPA Specification – Part 11, Agent Management support for Mobility*).

Un ejemplo es este mensaje con el que un agente solicita a su AMS la migración a una dirección remota.

**(request :sender** [**an-agent@async://fipa97.org/acc**](mailto:an-agent@async://fipa97.org/acc)

**:receiver an-ams@async://fipa98.org/acc :content (action an-ams@async://fipa98.org/acc (move (:agent-name an-agent@async://fipa97.org/acc) (:address async://fipa97.org/acc) (:destination an-ams@async://fipa98.org/acc) (:agent-profile ...) (:agent-mobility-protocol “...”) (:agent-code “...”) (:agent-data “...”) (:agent-version “...”) :protocol fipa-request :ontology fipa-mobile-agent-management )**

FIPA define dos modelos de movilidad en función de quién tenga que hacer la gestión de la transferencia:

**Modelo Simple**: El AMS (*Agent Management System*) es el responsable de realizar toda la gestión necesaria. El agente solicita a su AMS la transferencia y éste se ocupa de llevarla a cabo.

**Modelo Complejo**: El agente es el responsable de realizar toda la operativa que le permite desplazarse a la plataforma remota.

A gent P la tform A gent P la tform 4 . term inate(A ) 4 . term inate(A ) 2 . request(m ove A ) 2 . request(m ove A ) 3. e x ecu te (A) 3. e x ecu te (A) 1 . request(m ove A ) 1 . request(m ove A ) A A ’ Agent Platform 5. request (transfer A) | terminate(A’) 5. request (transfer A) | terminate(A’) 1. request(move A) 1. request(move A) 2. execute(A) 2. execute(A) 4. inform(move A) 4. inform(move A) A A’ Agent Platform 6. terminate(A) 6. terminate(A) 3. inform(okay | fail) 3. inform(okay | fail)

**9.3.- Problemas detectados**Conseguir la movilidad dentro de un sistema de agente no es nada sencillo.

Hay algunos problemas que todavía no han sido resueltos de una manera general, y hasta que no se consiga no podremos pensar en un sistema comercial que estén basado en agentes móviles como los que presenta la FIPA en su estándar.

Algunos de estos problemas se enumeran a continuación:

**Seguridad** tanto del agente como de la plataforma de agentes. Como ya se mencionó en el apartado dedicado a la seguridad, la movilidad de los agentes conlleva una cantidad importante de problemas de seguridad.

**Control** del estado de los agentes (centralizado vs distribuido). Un AMS controla en todo momento el estado y actividades de los agentes de su AP. Pero cuando los agentes pueden migrar, el modelo de gestión se complica. Si el mismo AMS debe gestionar el estado de los agentes desplazados debe haber algún tipo de intercambio de información para que este estado se vaya actualizando. Por otro lado un modelo centralizado supondría una entidad única de gestión se encargaría de controlar el estado de cada uno de los agentes de un sistema compuesto por varias plataformas.

**Integridad** del proceso de transferencia. La transferencia de los agentes debe ser una operación atómica, es decir, o se produce completamente o no se produce. Es importante asegurar que un agente ha alcanzado su destino antes de dar su transferencia por acabada. Esto tiene que ver mucho con el modelo transaccional de las bases de datos de gran importancia para mantener la integridad de la información.

**Continuidad** de las comunicaciones de los agentes móviles. Cuando un agente se desplaza va a perder todos los canales de información abiertos, ya sean ficheros, conexiones de red, etc. En el sistema remoto necesita retomar todos estos canales para seguir con su actividad.

**Requisitos de recursos** en cada lugar. Igual que otro producto software los agentes necesitan una serie de recursos mínimos para su ejecución (otros programas, librerías, clases, DLLs, componentes y recursos hardware). Cuando un agente se mueve a otro lugar debe asegurarse de que este nuevo sitio va a permitir su ejecución.

**Gestión de fallos**. Como en cualquier modelo distribuido la gestión de fallos es muy compleja, creciendo las posibilidades de incidencias de forma exponencial con la complejidad del sistema. La gestión de fallos es algo que todavía está muy lejos de ser una realidad en el mundo de los agentes.

**10.- Ontology Service**

Toda la plataforma FIPA se basa en la comunicación entre agentes mediante ACL (*Agent Communication Language*).

Como hemos visto un mensaje ACL consiste básicamente en una expresión de un determinado lenguaje y con un conjunto de términos específicos de la ontología usada.

Es decir, cuando dos agentes están conversando es necesario que ambos estén de acuerdo en el lenguaje y la ontología empleados.

Una Ontología es un conjunto de símbolos o términos junto su correspondiente interpretación o significado.

Cuando dos agentes usan una misma Ontología se aseguran que sus interpretaciones de lo que se está diciendo coinciden.

Cuando un agente se registra en el DF (*Directory Facilitator*) informa de las ontologías de las que tiene conocimiento, de tal forma que, cuando otro agente quiere conversar con él va a utilizar una de ellas consiguiendo un entendimiento mutuo.

En un dominio pequeño es posible que las ontologías aplicables se incluyan en el propio código del agente.

De tal forma que estas ontologías estarían cerradas y limitadas al contexto en el que van a ser utilizadas.

Cuando pensamos en sistemas de mayor envergadura no hay más remedio que utilizar Servicios de Ontología (*Ontology Agent*), de tal forma que todos los agentes puedan utilizar dichos servicios para obtener toda la información necesaria

Las labores básicas del Servicio de Ontología son:

Mantiene un conjunto de ontologías de uso público accesible a los agentes.

Traduce expresiones entre diferentes ontologías.

Responde a consultas sobre términos de las ontologías que gestiona.

Facilita la identificación y uso de ontologías compartidas entre los agentes.

Descubre nuevas ontologías y las pone a disposición de todos los agentes.

En la actualidad existen una serie de servidores de ontologías fuera del ámbito de FIPA, tal como, el servidor de Ontolingua , el servidor de XML/RDF y el servidor ODL a los que se accede a través de APIs como OKBC, HTTP y ODL respectivamente.

FIPA no pretende sustituir estas implementaciones sino agruparlas de forma que se puedan utilizar de forma transparente dentro de la plataforma. Como hemos dicho el *Ontology Agent* tiene como principal objetivo mostrar a los agentes FIPA una visión unificada de los servicios de ontología ofrecidos por los distintos servidores existentes.

OKBC Ontology Server OQL Ontology Server Ontology Agent (Ontolingua) (DB of ODL definitions) http Ontology Server ( XML ) ),3$ FRPSRQHQWV DF 2QWRORJ\ GHVLJQHU OA-2 1RQ\_),3$ FRPSRQHQWV $&&

**11.- Agent Message Transport**

**11.1.- Introducción**Es una de las pocas partes de estándar que habla más de diseño e implementación que de la propia especificación.

Como se ha mencionado anteriormente el APMT (*Agent Platform Message Transport*) es el encargado de realizar la transferencia de mensajes ACL entre las ACCs (*Agent Communication Channel*) de dos plataformas.

Aunque en un principio la labor de diseño de esta componente de bajo nivel se dejó en manos de cada grupo de trabajo que implementase la plataforma FIPA, finalmente se decidió introducir dentro del propio estándar debido a la gran importancia que tiene dentro del sistema

La figura siguiente muestra donde se ubica el AMT (*Agent Message Transport*) dentro de la platformas de agentes:

Agent Platform Agent Communication Channel Agent Platform Agent Communication Channel Agent Agent Message Transport Protocol ACL Message sent over the Message Transport Service

**11.2.- Modelo de transporte**

La figura anterior muestra de forma simplificada el modelo de objetos de la capa de transporte que propone FIPA en su estándar. Se recoge la comunicación extremo a extremo entre agentes, de forma que éstos se olvidan de cómo se produce realmente la comunicación y se preocupan de con quién se desean comunicar y qué es lo que desean comunicar. En esta arquitectura aparecen una serie de conceptos o elementos que es necesario conocer para poder comprender el modelo utilizado:

AMTS (*Agent Message Transport Service*): Es el objeto que implementa el servicio de transporte de mensajes en un AP. Cada agente de la plataforma puede conseguir una referencia al AMTS a través de AMS (*Agent Management System*).

*Conversation Factory* : Es una interfaz implementada por el AMTS que permite a un agente establecer un contexto de comunicación (*Conversation Context*). Por tanto cuando un agente desee empezar una conversación con otro necesita obtener un nuevo contexto.

*Conversation Context*: Representa un extremo lógico de un canal de comunicación establecido para el intercambio de mensajes ACL. Define la información que necesita mantener el sistema para lograr un flujo de mensajes entre dos agentes. Un agente tendrá tantos objetos de este tipo como número de conversaciones en el que esté implicado. De forma muy básica se puede decir que un contexto de comunicación está definido por el origen, el destino, un método de transporte y la calidad del servicio prestado. Además cada objeto de esta clase contendrá una cola de mensajes en espera de ser recibidos por el agente y una función de *callback* mediante la que se notifica al agente de la llegada de nuevos mensajes pertenecientes a dicha conversación.

*Transport Factory*: Es una interfaz implementada por el AMTS que permite a un agente investigar el conjunto de mecanismos de transporte soportados (iiop, http, wap,etc) y conocer algunas de las características de los mismos.

*Transport*: Representa un mecanismo de transporte de mensajes concreto. Cada vez que un agente desea establecer una nueva conversación tiene que identificar el método de transporte a usar o bien delegar esta responsabilidad al propio AMTS.

*Address*: Representa un extremo del canal de transporte utilizado. Dependiendo de del tipo de transporte elegido las direcciones serán diferentes (direcciones IP, URL, RFC\_822 email,etc).

*Destination*: Identifica la entidad remota con la que se está produciendo la comunicación o intercambio de mensajes. Puede haber un solo destino asociado con varias direcciones correspondientes a diferentes tipos de transporte. Este modelo se basa en la vida real en que una persona puede tener varias direcciones de contacto como son teléfono personal, teléfono móvil, fax, dirección postal, etc.

*Quality of Service*: Encapsula el conjunto de propiedades que gobiernan el modo en que el canal de transporte realizará el intercambio de mensajes entre los extremos de la comunicación. Algunas de estas propiedades son la integridad de datos, privacidad, rendimiento mínimo y ancho de banda mínimo. Esto permitirá elegir en cada momento la calidad de servicio que se desea contratar de acuerdo con las necesidades.

*Messag*e: Es la unidad básica de información que se va trasnmitir por el canal de comunicación. Como ya se ha explicado, esta unidad va a ser el mensaje ACL.

Una vez que se tienen todos estos componentes disponibles en el sistema el modelo de comunicación básico sería el siguiente:

1. Un agente A empieza su ejecución en una plataforma de agentes.

2. El agente A se registra como un participante potencial de conversaciones.

3. Un agente B empieza su ejecución en otra plataforma de agentes.

4. El agente B se registra como un participante potencial de conversaciones.

5. El agente B identifica a A como un agente capaz de conversar.

6. El agente B crea un contexto de comunicación para conversar con A.

7. El agente B envía un mensaje al agente A.

8. El AMTS de la plataforma de agentes de A recibe el mensaje enviado por B.

9. Un contexto de comunicación es establecido por A para comunicarse con B.

10. El mensaje de B es enviado a A mediante la función de callback establecida.

11. El agente A contesta al mensaje del agente B.

12. Se produce un intercambio de mensajes.

13. El agente A finaliza su comunicación con el agente B.

14. El agente B es notificado de la finalización de la comunicación

**12.- Agent Naming**

En un principio FIPA fijó un modelo de nombrado de agentes estático donde cada agente tenía un identificador único (GUID) que tenía la siguiente estructura:

***<name>@<hostname>:<port>/<target>***

Name : identificador del agente en la plataforma de agentes a la que pertenece.

HostName: Dirección IP (o nombre DNS) del host en el cual está corriendo su ACC.

Port: Puerto en el que escucha el ACC.

Target: Identificador del agente que debería recibir el mensaje para enrutarlo al agente destino. Normalmente será el propio ACC.

Un ejemplo de GUID podría ser: [*acc@iiop://www.ucm.es:50/acc*](mailto:acc@iiop://www.ucm.es:50/acc)

Posteriormente se decidió que la estructura del nombre debía ser más flexible y completa para acometer futuras necesidades.

**AgentName= (“:guid” Word NameFields\*) NameFields= “:addresses” “(“ CommAddress+ “)” | “:resolver” “(“ AgetName+ “)” | “:authenticator” “(“ AgentName+ “)”**

GUID: Identificador único del agente en la plataforma de agentes. No tiene porque contener la dirección de transporte como en el modelo simple.

Addresses: Dirección/es de transporte en la que se encuentra el agente.

Resolver: Dirección donde se encuentra el Servidor de Nombres (ahora en el AMS) Authenticator:

Dirección del agente de autentificación (ahora el propio AMS).

Ejem: *:guid MiAgente@iiop://www.ucm.es :resolver ( :guid ams@www.ucm.es :addresses (iiop://www.ucm.es:81/ams http://www.ucm.es:80/ams) ) :addresses (iiop://fipa.org/MiAgente) :authenticator (iiop://fipa.org/Autentificación)*

Este modelo que está siendo desarrollado por FIPA está basado en pares atributo-valor y por tanto puede ser extendido de forma sencilla para recoger nuevas necesidades que pudiesen aparecer en el futuro.

**13.- Términos y definiciones**

**Acción (*action*):**Un elemento básico que representa una actividad que un agente puede realizar. Una clase especial de acción es un acto de comunicación ( communicative act o speech act).

**Agente ARB (*ARB Agent*):**Un 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).

**Agente (Agent):**Es una entidad software que combina uno o más servicios o capacidades en un modelo de ejecución integrado y autónomo. Es el elemento básico de un sistema multiagente.

**Agent Communication Language (ACL):**Un lenguaje con una sintasis, semántica y pragmática formalmente definidas. Es la base de las comunicaciones entre agentes de una naturaleza hetereogénea.

**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 de 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 del agente y la dirección de transporte donde se encuentra.

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

**Autonomía :**Capacidad de una entidad de realizar acciones de acuerdo con sus propias decisiones.

**Clonación :**Proceso por el cual un agente es copiado dentro de una plataforma de agentes.

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

**Data Mining (Minería de datos):**Conjunto de técnicas utilizadas para poder obtener la información relevante a partir de un conjunto maxivo de datos difíciles de analizar.

**Data Warehouse (Almacen de datos):**Almacen de datos que contiene todos los datos relevantes sobre los indicadores de una empresa. Estos datos contienen gran información pero es difícil obtenerla y explotarla.

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

**Divide y Venceras:**Estrategía de resolución de problemas que consiste en descomponer el problema en problemas más sencillo, resolver estos por separado y a partir de sus soluciones alcanzar la solución del problema original.

**Information Age (Era de la Información) :**Denominación que recibe la sociedad que está naciendo en la que las teconologías de la información son cada vez más importantes. El nacimiento de esta nueva era llegó con la aparición de Internet y la WWW ( World Wide Web).

**Knowledge Querying and Manipulation Language (KQML):**Un estándar de facto, ampliamente utilizado, que proporciona una especificación de un lenguaje para la comunicación entre agentes.

**Metabuscador :**Buscador de infromación en Internet basado en la utilización lo mejor de otro motores de búsqueda para conseguir unos mejores resultados. Normalmente se guarda información adicional para poder realizar búsquedas más eficientes.

**Migración:**Proceso por el cual un agente se traslada a otro nodo de la red para continuar su ejecución.

**Agente Móvil (Mobile agent) :**Un agente que puede desplazarse desde una plataforma de agentes hasta otra y seguir su ejecución en el mismo punto donde lo dejó.

**Nomadic Application:**Una aplicación distribuida que da servicios a usuarios de dispositivos móviles como portátiles, PDAs y teléfonos móviles.

**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 intercambiado.

**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 dialogo entre agentes.

**TCP/IP:**Un protocolo de red usado para establecer conexiones y transmitir datos entre máquina. Es el protocolo de usado en Internet.

**User Dialog Management Service (UDMS):**Un servico que permite a una plataforma de agentes interaccionar con un usuario humano. Su labor se convertir los mensajes ACL en un formatos compresible por el usuario (GUI, voz, etc. ) y viceversa.

**User Personalization Service (UPS):**Un servicio que permite mantener el modelo de los usuarios que interaccionar con el sistema mediante el UDMS.

**Modelo de Usuario (User Model):**Un modelo de usuario contiene información acerca de las preferencias, capacidades, objetivos, conocimiento, etc. de un usuario que interacciona con el sistema. Esta información puede ser obtenida por un proceso de deducción basado en el comportamiento del usuario.

**Reactiva:**Propiedad de una entidad que es capaz de detectar estímulo externos y actuar de acuerdo con ellos.

**Social :**Propiedad de una entidad que es capaz de colaborar con otras entidades para alcanzar soluciones comunes.

**WAP:**Wireless Application Protocol es un arquitectura para el desarrollo de aplicaciones en sistemas inalámbricos desarrollada por un consorcio de industrias denominado WAP Forum. La arquitectura está basada en un modelo cliente/proxy/servidor, y se creo para resolver los problemas que aparecen cuando se aplica Internet a sistemas con muy poco ancho de banda o con poca capacidad de procesamiento.

**Wireless Environment:**Un entorno de comunicación en donde al menos uno de los componentes usados se basa en señales de radio.

**Wrapper Agent:**Un agente que recubre a un componente software para proporcionar una vista del mismo similar a la de cualquier agente FIPA. Esto permite que dicho componente sea utilizado dentro de una plataform de agentes.

**14.- Simbolos y términos abreviados**

ACC: Agent Communication Channel

ACL: Agent Communication Language

AMS: Agent Management System

AMTS: Agent Message Transport service

ANS: Agent Name Service

P: Agent Platform

API: Application Programming Interface

APMT: Agent Platform Message Transport

ARPA: Advanced Research Projects Agency.

CORBA: Common Object Request Broker Architecture

DB: Database

DF: Directory Facilitator

FIPA: Foundation for Intelligent Physical Agents

GUI: Graphic User Interface

GUID: Global Unique Identifier

HTTP: Hypertext Transmission Protocol

IA: Inteligencia Artificial

IDL: Interface Definition Language

IIOP: Internet Inter-ORB Protocol

IPMT: Internal Platform Message Transport

IT: Information Technology

KIF: Knowledge Interchange Format

KQML: Knowledge Querying and Manipulation Language

KSE: Knowledge Sharing Effort

MAF: Mobile Agent Facility

OMG: Object Management Group

ORB: Object Request Broker

RMI: Remote Method Invocation, an inter-process communication method embodied in Java

SL: Semantic Language

SMTP: Simple Mail Transfer Protocol

TCP / IP: Transmission Control Protocol / Internet Protocol

UDMA: User Dialogue Management Agent

UDMS: User Dialogue Management Service

UPA: User Personalization Agent

UPS: User Personalization Service

VPN: Virtual Private Network.

XML: eXtensible Markup Language

WAP: Wireless Application Protocol

**15.- Bibliografía**

FIPA Specification \_ <http://www.fipa.org>

Intelligent Software on the Internet \_ <http://www.hermans.org/agents>

Agentes Móviles \_ *Juan Pavón Mestras, SEID’99*

Agentes: El Mercado Virtual \_ *Rev. e.comm, Octubre 99*

Agentes \_ <http://www.cetus-links.org>

Agent Society Home Page \_ <http://www.agent.org>

OMG Home Page \_ <http://www.omg.org>

Intelligent Agent: Theory and Practice. \_ *Knowlenge Engineering Review, June 1995*\_ <http://gryphon.elec.qmw.ac.uk/dai/pubs/KER95/>

RPP. Revista Profesional para programadores, Año V/ Nº 10. \_ *¿Qué es el Data Warehousing y Data Mining?*

White Paper from Royal Melbourne Institute of Technology*.* Elizabeth A. Kendall*.*\_ *The Layered Agent Patterns.A Java Application Framework for Agent Based Systems*

Mobile Agent Facility Specification (MASIF) \_ http://www.omg.org