

Curso de doctorado: Agentes Software

Juan Pavón Mestras
Dep. Sistemas Informáticos y Programación
Universidad Complutense Madrid



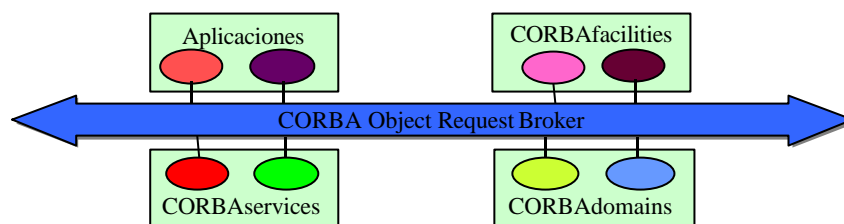
Estandarización de sistemas de agentes

Organizaciones de estandarización

- **OMG (Object Management Group)**
 - Mobile Agent System Interoperability Facilities (MASIF)
- **KSE (Knowledge Sharing Effort)**
 - Knowledge Querying and Manipulation Language (KQML)
 - Knowledge Interchange Format (KIF)
- **FIPA (Foundation for Intelligent Physical Agents)**
 - Especificaciones de arquitectura, infraestructura y aplicaciones
- **Agent Society**
 - Arquitectura y protocolos de comunicación genéricos

OMG

- Object Management Group (OMG): <http://www.omg.org>
 - Asociación de más de 800 empresas/instituciones formado en 1989
 - Objetivos:
 - INTEROPERABILIDAD
 - Reusabilidad
 - Portabilidadde sistemas distribuidos de componentes de software orientado a objetos
 - CORBA

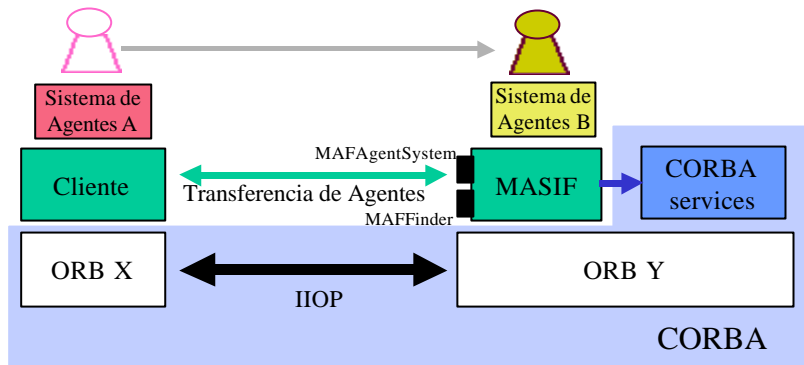


MASIF

- Considera los AM como objetos CORBA que tienen la posibilidad de moverse, ejecutarse autónoma y asíncronamente en sistemas de ejecución seguros (sistemas de agentes)
- MASIF define cómo CORBA puede soportar:
 - movilidad de agentes
 - invocación de sistemas de agentes
- utilizando varios CORBAservices:
 - Security
 - Naming
 - Lifecycle
 - Externalization

MASIF

AM en la arquitectura CORBA



MASIF

- Aspectos estandarizados
 - Gestión de agentes
 - Crear, arrancar, suspender, terminar un agente
 - Transferencia de agentes
 - Nombres de agentes y de sistemas de agentes
 - Tipos de sistemas de agentes y sintaxis de localización
 - Identificación de sistemas de agentes
 - Autoridad de nombres
- Estandarización futura
 - Comunicación ente agentes
 - Seguridad multi-salto
 - Puentes entre distintos tipos de sistemas de agentes
 - Cómo ejecutar agentes sobre distintos tipos de sistemas de agentes

MASIF

- 2 interfaces a nivel de sistema de agentes:
 - *MAFAgentSystem*
 - Operaciones de agente: crear, recibir, suspender, terminar, obtener información de estado
 - Operaciones sobre autoridad
 - Operaciones sobre el sistema de agentes: lugares y agentes existentes, vecinos
 - *MAFFinder*: Operaciones para
 - registrar
 - desregistrar
 - localizaragentes, lugares, y sistemas de agentes
- No hay interfaz unificada para los agentes
 - los agentes sólo viajan a sistemas de agentes que soporten su perfil

MASIF

Nombrado

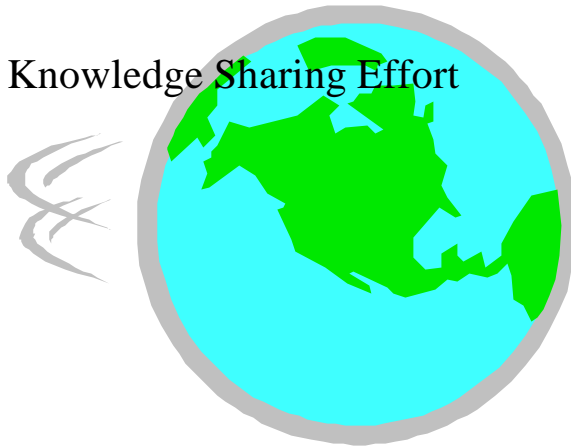
```
typedef short AgentSystemType;

typedef sequence<octet> OctetString;
struct ClassName {
    string name;
    OctetString discriminator;
}
typedef sequence<ClassName> ClassNameList;
typedef sequence<OctetString> OctetStrings;
typedef OctetString Authority;
typedef OctetString Identity;

struct Name {
    Authority          authority;
    Identity           identity;
    AgentSystemType   agent_system_type;
};
typedef string Location;
```

KSE

Knowledge Sharing Effort



KSE

Knowledge Sharing Effort

- Iniciado por ARPA hacia 1990, y apoyado por organismos norteamericanos de investigación (ASOFR, NSF, NRI)
- Propósito:
 - Desarrollo de técnicas, metodologías y herramientas software para la compartición y reutilización del conocimiento entre sistemas
 - ... a lo largo de las etapas del ciclo de vida del software:
 - diseño
 - implementación
 - ejecución

KSE

Knowledge Sharing Effort

- Compartir conocimiento entre agentes requiere la capacidad de comunicarse
- La capacidad de comunicarse requiere un lenguaje de comunicación común:
 - Sintaxis: *KIF* (Knowledge Interchange Format)
 - Semántica: *Ontolingua* (lenguaje para definir ontologías)
 - Pragmática: *KQML* (Knowledge Query and Manipulation Language)

KIF

- Aspectos sintácticos de la representación del conocimiento para expresar
 - Conocimiento
 - Meta-conocimiento
- basado en cálculo de predicados de primer orden con extensiones para soportar:
 - razonamiento
 - definiciones
- Especificación actual en <http://logic.stanford.edu>

KIF

- Ejemplos de expresiones
 - Datos
(empleado pepe ventas)
tupla que indica que pepe es un empleado en el departamento de ventas
 - Operaciones lógicas
(> (* (ancho chip1) (largo chip1)) (* (ancho chip2) (largo chip2)))
establece que un chip es de mayor tamaño que el otro
 - Conocimiento sobre conocimiento (usando los operadores ‘ y ,)
(interesado pepe ‘(empleado ,?x ,?y))
el agente pepe está interesado en recibir tuplas de la relación de empleados
 - Procedimientos (con sintaxis parecida a LISP)
(progn (frase-tipica f)
 (print “Hola mundo.”)
 (frase-tipica f))

Ontolingua

- Ontología: un vocabulario común en el que se han acordado significados para describir un dominio
 - Es una conceptualización del mundo, en función de objetos, cualidades, distinciones, relaciones
- Un agente satisface una ontología si sus acciones observables son consistentes con las definiciones de la ontología
- Las ontologías se describen en KIF usando el vocabulario de definición Ontolingua
 - Una ontología define un conjunto de clases, funciones y constantes para un dominio de discurso, e incluye una axiomatización para restringir su interpretación

Ontolingua

- Ontolingua es un lenguaje que permite construir, publicar y compartir ontologías
 - Una interfaz a un servidor de edición/navegación
 - Las ontologías pueden traducirse automáticamente a distintos lenguajes de contenidos, como KIF, LOOM, Prolog, CLIPS, etc.
 - El lenguaje incluye primitivas para combinar ontologías
- Ejemplos de ontologías:
 - Cyc, ontología de propósito general
 - WordNet, sistema de referencia léxica accesible por internet
 - World Fact Book, 5 megas de KIF...
 - UMLS, sobre medicina

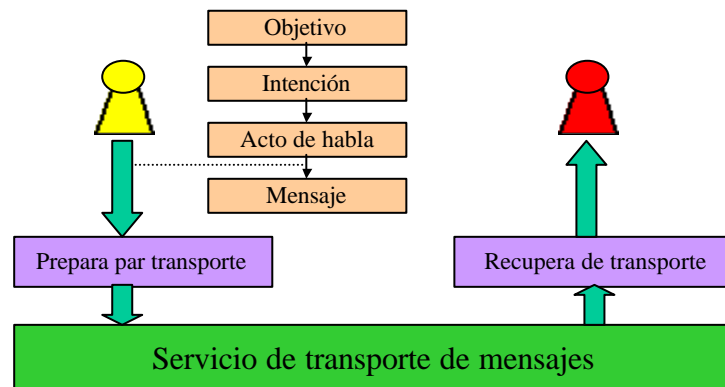
KQML

- Es un lenguaje de comunicación y protocolo, orientado a mensajes, para el intercambio de información
- KQML es independiente de
 - protocolos de transporte (TCP/IP, HTTP, ...)
 - sintaxis de contexto
 - ontologías
 - protocolos de alto nivel (contract net, subasta, ...)

KQML

- Asume un modelo de agentes:
 - entidades de alto nivel con capacidades cognitivas (representación simbólica, base de conocimientos, ...)
 - tienen una descripción de nivel intencional: su estado es un conjunto de componentes mentales como creencias, capacidades, elecciones, compromisos, etc.
 - ☞ los agentes residen en el *nivel del conocimiento*
- Los mensajes KQML comunican una actitud sobre el contenido que llevan (aserto, solicitud, pregunta)
 - Las primitivas del lenguaje se llaman *performativas*
 - Concepto de la teoría del acto del habla (*speech act*)

KQML



KQML

- Nivel de contenidos
 - Lleva el contenido del mensaje en el lenguaje de representación propio de los agentes
 - No es procesado por las implementaciones de KQML (se deja al agente)
- Nivel de mensajes
 - Núcleo del lenguaje KQML, determina los tipos de interacción que pueden tenerse con un agente que hable KQML
 - Identifica el protocolo que se utiliza para enviar un mensaje y proporciona una performativa al emisor que añade el contenido
 - Indica también el lenguaje de contenidos, ontología asumida, y algún descriptor sobre el contenido (esto permite a algunas aplicaciones procesar los mensajes sin acceder al contenido)
- Nivel de comunicaciones
 - Trata los parámetros de comunicación de más bajo nivel: identidad de emisor y receptor, identificación de la comunicación

KQML

- Mensajes KQML
 - Representa un acto de habla o performativa
 - Consta de una lista de pares atributo-valor

(ask-one ← performativa nivel de contenidos
:sender pepe
:content (PRECIO TELEFONICA ?precio)
:receiver servidor-bolsa ← nivel de comunicaciones
:reply-with accion-telefonica
:language LPROLOG ← nivel de mensajes
:ontology IBEX
)

KQML

- Mensajes KQML

La respuesta al anterior:

```
(tell
  :sender servidor-bolsa
  :content (PRECIO TELEFONICA 19)
  :receiver pepe
  :in-reply-to accion-telefonica
  :language LPROLOG
  :ontology IBEX
)
```

KQML

- Performativas más comunes

- ❑ Preguntas: *evaluate, ask-if, ask-about, ask-one, ask-all*
- ❑ Respuestas: *reply, sorry*
- ❑ Multirespuestas: *stream-about, stream-all, eos*
- ❑ Información: *tell, achieve, cancel, untell, unachieve*
- ❑ Generador: *standby, ready, next, rest, discard, generator*
- ❑ Definición de capacidades: *advertise, subscribe, monitor, import, export*
- ❑ Networking: *register, unregister, forward, broadcast, route*

KQML

- Semántica de las performativas (Labrou y Finin)

- Ejemplo: tell

tell (A, B, X)

Pre(A): Bel(A,X) \wedge Know(A, Want(B, Know(B, S)))

Pre(B): Int(B, Know(B, S))

siendo S: Bel(B, X) \vee \neg Bel(B,X)

Post(A): Know(A, Know(B, Bel(A,X)))

Post(B): Know(B, Bel(A,X))

Completion: Know(B, Bel(A,X))

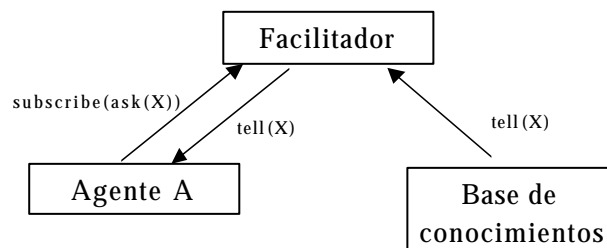
- Un agente no puede ofrecer a otro agente información no solicitada
- Obsérvese que en la semántica no se indica cómo el agente A llega a saber que el agente B quiere saber esa información (puede ser por petición de B o porque un tercero se lo haya solicitado a A)

KQML

Facilitadores y mediadores con KQML

- Agentes facilitadores

- Asociación de nombres simbólicos con direcciones físicas
 - Registro de bases de datos y servicios ofrecidos y buscados por los agentes
 - Servicios de comunicaciones (reenvío, brokering, etc.)



El agente A puede solicitar al facilitador que monitorice cambios en la base de conocimientos

KQML

- Actualmente existen varios dialectos de KQML con distintas extensiones:
 - añadiendo performativas
 - offer, accept, ...
 - añadiendo nuevos parámetros
 - :protocol, :version, :signature, ...
 - aspectos de seguridad
- Plataformas de agentes basadas en KQML y Java:
 - Java Agent Template, Lite (*JATLite*), Stanford Univ.
 - Java-based Agent Framework for Multi-Agent Systems (*JAFMAS*), Univ. of Cincinnati
 - *Jackal*, Univ. of Maryland

FIPA



FIPA

- Foundation for Intelligent Physical Agents
 - Consorcio industrial fundado en 1996
 - Varias decenas de compañías de telecomunicaciones e informática
- ☞ <http://www.fipa.org>
- Objetivos
 - Acelerar el desarrollo de tecnologías de agentes inteligentes mediante la producción de especificaciones acordadas internacionalmente
 - Especificación del comportamiento y capacidades externas de subsistemas genéricos: recursos de agentes (para migración, ejecución, etc.), interacción y cognitivos
 - las especificaciones estarán basadas en casos prácticos concretos
 - Agentes, multi-agentes, y sociedades de agentes
 - Selección y adaptación de tecnologías existentes

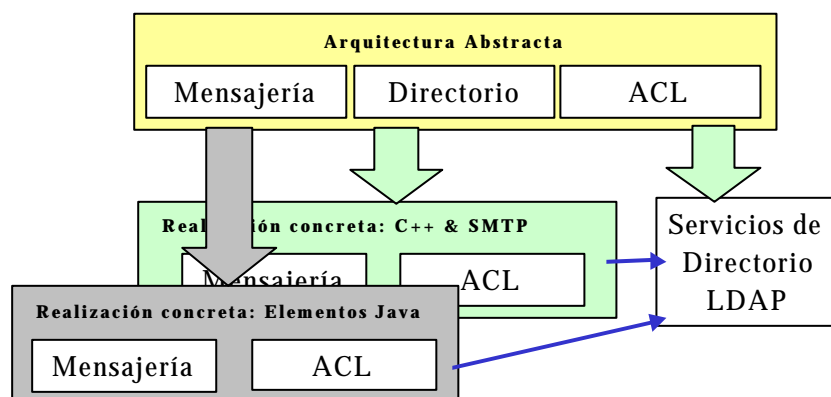
FIPA

- Especificaciones FIPA 97 y FIPA 98
 - Tecnología de agentes
 - Gestión de agentes
 - Lenguaje de comunicación entre agentes
 - Interacción agente-humano
 - Interacción de agentes con sistemas software
 - Movilidad de agentes
 - Seguridad de agentes
 - Aplicaciones de prueba
 - Agente personal de viaje
 - Asistente personal
 - Difusión y entretenimiento audio-visual
 - Gestión de redes

FIPA

- FIPA 2000
 - Evolución del modelo de comunicaciones de agentes:
 - Transporte de los mensajes
 - Representación de los mensajes (como strings, objetos, o XML)
 - Atributos opcionales en los mensajes (p.ej. autenticación, encriptado)
 - Uso de entornos software existentes:
 - Plataformas y lenguajes de programación para computación distribuida
 - Plataformas de mensajería
 - Servicios de seguridad
 - Servicios de directorio
 - Tecnologías de conectividad intermitente (móviles)
- ☞ **Arquitectura Abstracta**
 - Define cómo dos agentes se pueden localizar y comunicar

Arquitectura Abstracta de FIPA



Arquitectura Abstracta de FIPA

- Los agentes se comunican intercambiando mensajes
 - que representan actos de habla
 - codificados en un lenguaje de comunicación de agentes
- Servicios de soporte a los agentes:
 - Servicios de directorio
 - Servicios de transporte de mensajes
- Los servicios se pueden implementar
 - como agentes
 - como software que se accede invocando métodos: C++, Java, IDL

Arquitectura Abstracta de FIPA

Servicios de Directorio

- Soporta un conjunto de entradas formadas por varias tuplas cada una con dos pares clave-valor:
 - Nombre de agente: único globalmente
 - Localizador: Uno o más descriptores de transporte que describen el tipo de transporte y la dirección de transporte específica para comunicarse con el agente
- Los agentes *registran* entradas de directorio para que otros agentes puedan *buscarlas* con el propósito de encontrarles para poder interactuar
- Ejemplo de entrada de directorio:

Nombre de agente: AgenteEjemplo

Localizador: *tipo de transporte* *dirección de transporte*

HTTP http://www.agentes.com/AgenteEjemplo

SMTP AgenteEjemplo @agentes.com

Atributos del agente: *Idiomas:* English, Español

Negociación preferida: contract-net

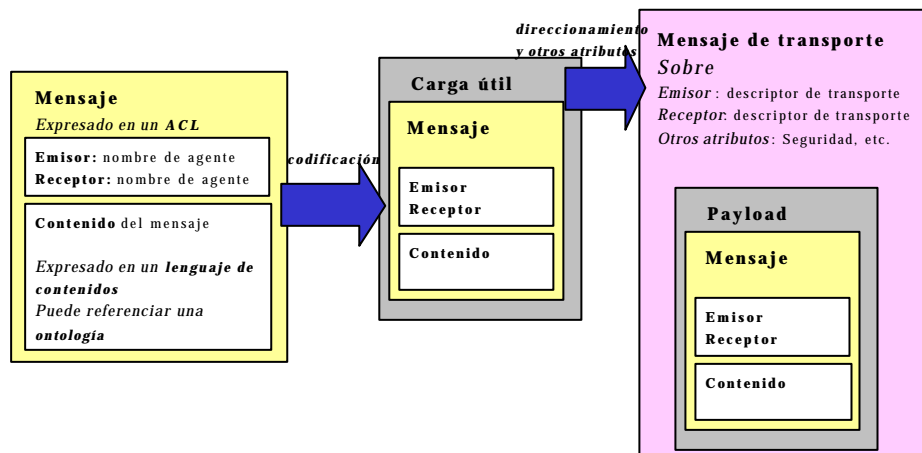
Arquitectura Abstracta de FIPA

Mensajes de los agentes

- Estructura de los mensajes
 - Los mensajes son tuplas clave-valor
 - Escritos en un lenguaje de comunicación de agentes (p.ej. FIPA ACL)
 - El contenido expresado con un lenguaje de contenidos (KIF, SL, ...)
 - El lenguaje de contenidos puede hacer referencia a una ontología
 - Incluyen los nombres de emisor y receptor
 - Un mensaje puede contener recursivamente otros mensajes
- Transporte de los mensajes
 - El mensaje de transporte consta de un campo de carga útil (*payload*) y un sobre (*envelope*)
 - El mensaje se transporta en la carga útil de un mensaje de transporte

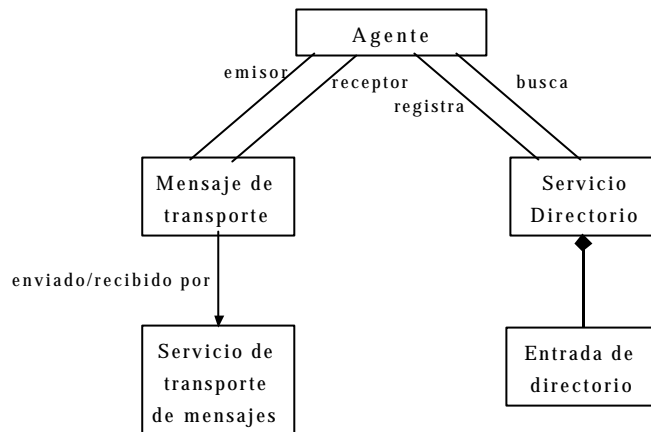
Arquitectura Abstracta de FIPA

Mensajes de los agentes



Arquitectura Abstracta de FIPA

Relaciones básicas del agente

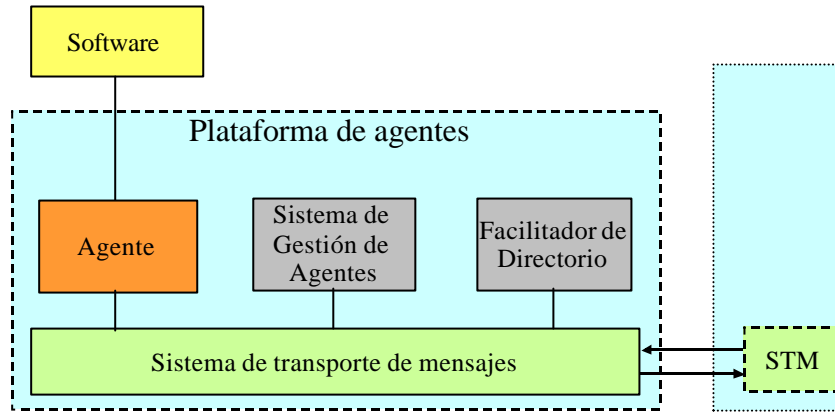


Gestión de agentes FIPA

- Define el modelo de referencia lógico para la creación, registro, localización, comunicación, migración y retirada de agentes
- El modelo de referencia define la *plataforma de agentes*:
 - *Sistema de gestión de agentes*: supervisa el acceso y uso de la plataforma de agentes
 - *Facilitador de directorio*: servicio de páginas amarillas
 - *Sistema de Transporte de Mensajes*: comunicación entre agentes de distintas plataformas de agentes

Gestión de agentes FIPA

Modelo de referencia



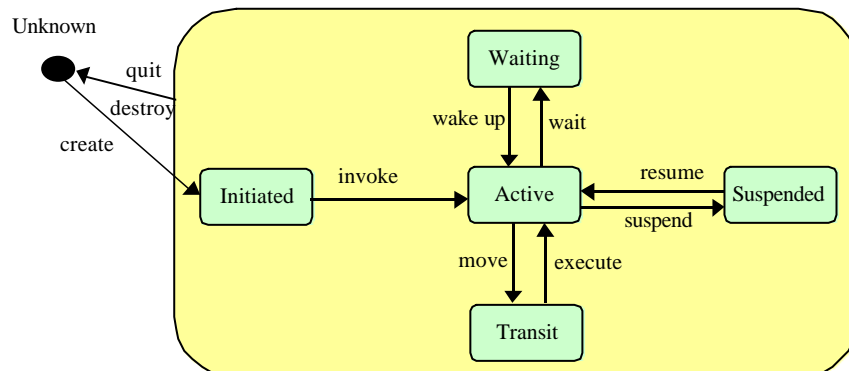
© JPM, UCM 2001

Estandarización de Sistemas de Agentes

37

Gestión de agentes FIPA

Ciclo de vida de un agente



© JPM, UCM 2001

Estandarización de Sistemas de Agentes

38

FIPA ACL

- Basado en actos del habla
 - La semántica se basa en aptitudes mentales (creencias, intenciones, etc.)
- Los mensajes son acciones comunicativas
 - Sintaxis similar a KQML
 - Semántica formal definida con lógica modal
- Además se definen protocolos de interacción de alto nivel, llamados *conversaciones*
- Es posible definir nuevas primitivas a partir de un núcleo de primitivas mediante composición
- El estado de los agentes se describe con el lenguaje SL (Semantic Language)

FIPA ACL

- Ejemplo de mensaje ACL

```
( inform
  :sender agente1
  :receiver agente2
  :content
    (precio libro 1000)
  :in-reply-to round-4
  :reply-with bid04
  :language sl
  :ontology ontolibro
)
```

FIPA ACL

Parámetros del mensaje

:sender	:language
:receiver	:ontology
:content	:reply_by
:reply_with	:protocol
:in_reply_to	:conversation-id
:envelope	

Tipos de mensaje

accept-proposal	inform	refuse
agree	inform-if	reject-proposal
cancel	inform-ref	request
cfp	not-understood	request-when
confirm	propose	request-whenever
disconfirm	query-if	subscribe
failure	query-ref	

FIPA ACL

SL (Semantic Language)

- Basado en Cohen y Levesque '90, y en Sadek '92
- Es una lógica multimodal con operadores modales para creencias (B), deseos (D), creencias falsas (U), e intenciones u objetivos persistentes (PG)
- Con SL se pueden representar
 - Propositiones
 - Objetos
 - Acciones

FIPA ACL

SL (Semantic Language)

- La semántica de cada acto comunicativo de FIPA ACL se define como un conjunto de fórmulas SL que describen:
 - Las precondiciones de factibilidad (FP): las condiciones necesarias para el emisor del acto comunicativo
 - aunque se den las condiciones el agente no está obligado a realizar el acto comunicativo, puede elegir
 - El efecto racional (RE): lo que un agente espera que ocurra como resultado de realizar la acción
 - pero el agente receptor no está obligado a asegurar que ocurra el efecto esperado ya que incluso puede resultarle imposible lograrlo

FIPA ACL

SL (Semantic Language)

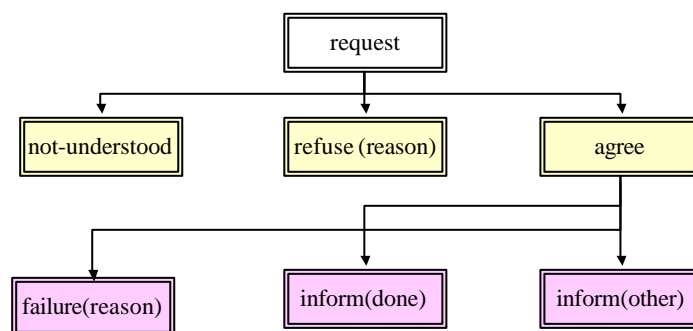
- Ejemplo: Semántica del acto comunicativo *inform*
 - El agente a informa al agente b del contenido X
 - $\langle a, \text{inform}(b, X) \rangle$
 - FP:** $B(a, X) \wedge \neg B(a, B(a, f(b, X))) \vee U(a, f(b, X))$
 - RE:** $B(b, X)$
 - El agente a cree X
 - pero no cree todavía que el receptor tenga algún conocimiento sobre la verdad de la proposición X
 - así que tiene la intención de que el agente receptor debería también llegar a creer que la proposición X es cierta (efecto racional $B(b, X)$)
 - Diferencia con *tell* en KQML, es que $\langle a, \text{inform}(b, B(a, X)) \rangle$

FIPA ACL

- Las conversaciones entre agentes suelen seguir unos ciertos patrones, secuencia típicas de mensajes: **protocolos de conversación**
 - Un agente informa del protocolo que quiere usar mediante el parámetro **:protocol**
- Protocolos básicos definidos por FIPA:
 - FIPA-request
 - FIPA-query
 - FIPA-request-when
 - FIPA-contract-net
 - FIPA-iteraterated-contract-net
 - FIPA-auction-english
 - FIPA-auction-ducth

FIPA ACL

Protocolo FIPA-request

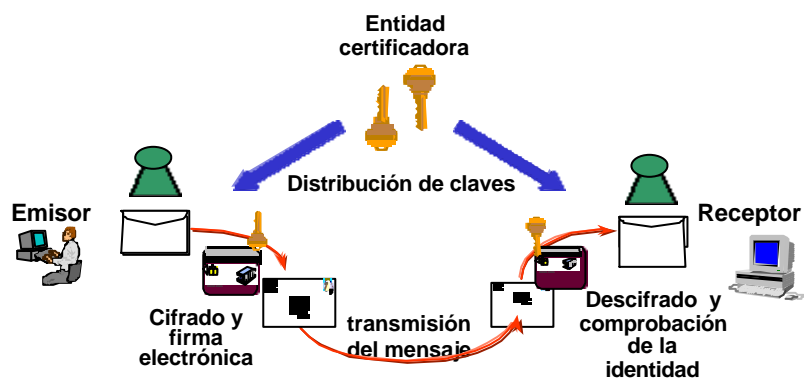


FIPA ACL

- Comparación con KQML:
 - Sintácticamente son muy similares
 - Ningún lenguaje se compromete con un solo lenguaje de contenidos
 - Se diferencian básicamente en la interpretación semántica de las performativas:
 - En la descripción:
 - KQML utiliza pre- y post- condiciones
 - FIPA ACL usa precondiciones de factibilidad (FP, feasibility preconditions) y efecto racional (RE, rational effect)
 - En el lenguaje para describir las actitudes proposionales:
 - El operador Bel en KQML no es el mismo que el operador B de FIPA
 - FIPA ACL es más potente en la composición de nuevas primitivas

Seguridad de agentes

- Sistema basado en la infraestructura de clave pública



Seguridad de agentes

- Por cada entidad, la autoridad certificadora 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 usuario (entidad) recibe sus claves privadas por parte de la autoridad certificadora. Las claves públicas de todas las entidades están guardadas por la entidad certificadora en una estructura de directorio (que admite búsquedas LDAP) y que permite la consulta de las mismas
- Cuando el emisor E envía un mensaje al receptor R:
 - ❑ Lo cifra con la clave pública del R, de este modo el contenido del mensaje solo podrá ser entendido 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, así cuando R reciba el mensaje podrá comprobar que realmente ha sido enviado por E (usando la clave pública de firmado de E para verificar la firma).

Movilidad de agentes

- Protocolos
 - ❑ Migración.
 - El código del agente y sus datos se mueven físicamente por la red
 - ❑ Clonación
 - Una copia del agente es enviada al nodo remoto
 - ❑ Invocación
 - Se solicita la creación de un agente en el nodo remoto
- Operaciones (mensajes ACL)
 - ❑ move
 - El agente se envía a la plataforma de agentes destino
 - ❑ transfer
 - Se transfiere la identidad y certificación a otro agente (modelo complejo)
 - ❑ execute
 - Se reanuda la ejecución del agente

Movilidad de agentes

Move:

```
(request
:sender 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)
```

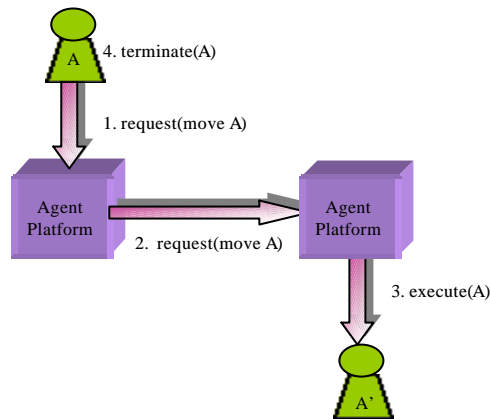
Movilidad de agentes

Transfer:

```
(inform
:sender an-agent@async://fipa97.org/acc
:receiver an-agent@async://fipa98.org/acc
:content
(action an-agent@async://fipa98.org/acc
(transfer
(agent-name an-agent@async://fipa98.org/acc)
(signature "..."))
:protocol fipa-request
:ontology fipa-mobile-agent-management)
```

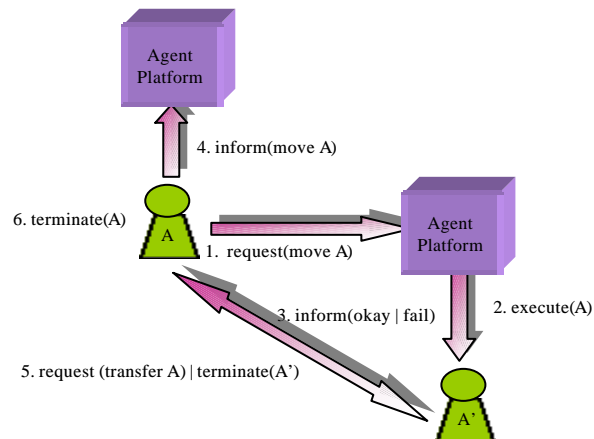
Movilidad de agentes

Protocolo de migración simple:



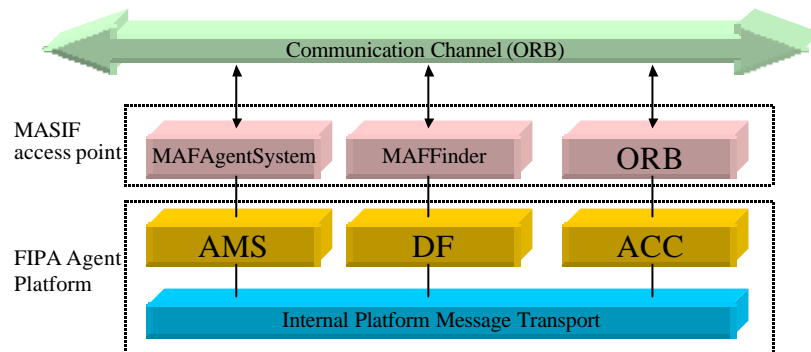
Movilidad de agentes

Protocolo de migración complejo:



Movilidad de agentes

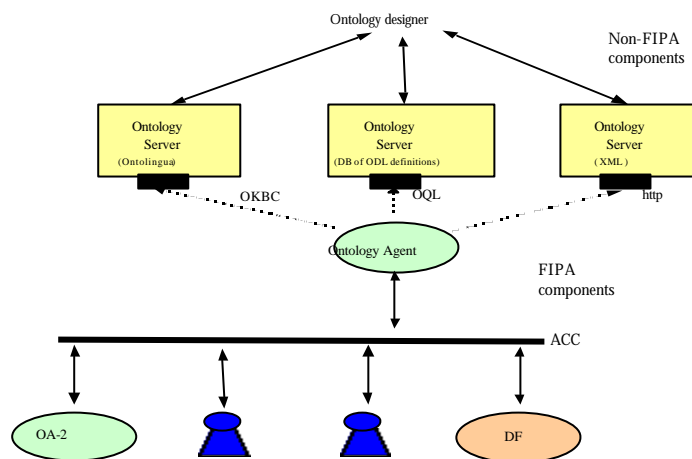
Integración con MASIF:



Servicio de ontología

- Definición
 - El modelo de comunicación FIPA se basa en que dos agentes que conversan comparten una ontología común para el dominio del discurso.
 - Una ontología es un conjunto de símbolos(constantes) asociados con su correspondiente interpretación o significado.
 - En un dominio dado se puede usar una ontología publicada por un servicio de ontología (Ontology Agent), o bien se puede incluir dicha ontología en el código del propio agente.
 - Ontology Agent (OA):
 - Descubre ontologías públicas para ponerla a disposición de los agentes del sistema.
 - Mantiene un conjunto de ontologías.
 - Traduce expresiones con diferentes ontologías.
 - Responde a consultas sobre terminos de la ontología.
 - Facilita la identificación de ontologías compartidas entre dos agentes.

Servicio de ontología



Estandarización de Sistemas de Agentes

57

Nombrado de agentes

Modelo:

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

GUID: Identificador único de un agente

- Name*: Identificador del agente en la plataforma de agentes.
- HostName*: Dirección IP (o nombre DNS) del host en el cual está corriendo un ACC.
- Port*: Puerto en el que escucha el ACC para nuevas peticiones.
- Target*: Es el identificador del agente que debería recibir el mensaje para enrutarlo al agente destino. Normalmente será el propio ACC.

Ejemplo: `acc@iiop://www.ucm.es:50/acc`

© JPM, UCM 2001

Estandarización de Sistemas de Agentes

58

Nombrado de agentes

Modelo avanzado

```
AgentName =      (":guid" Word NameFields* )
NameFields =    ":addresses" "(" CommAddress+ ")"
                /      ":resolver" "(" AgentName+ ")"
                /      ":authenticator" "(" AgentName+ ")"
```

Agent Name:

- GUID*: Identificador único del agente en la plataforma de agentes (no tiene porque contener la dirección de transporte como en el modelo anterior).
- Addresses*: Dirección/es de transporte en la que se encuentra el agente.
- Resolver*: Dirección donde se encuentra el *White Pages Service* (ahora en el propio AMS).
- Authenticator*: Dirección de agente de autenticación (ahora en el propio AMS).

Objetivos:

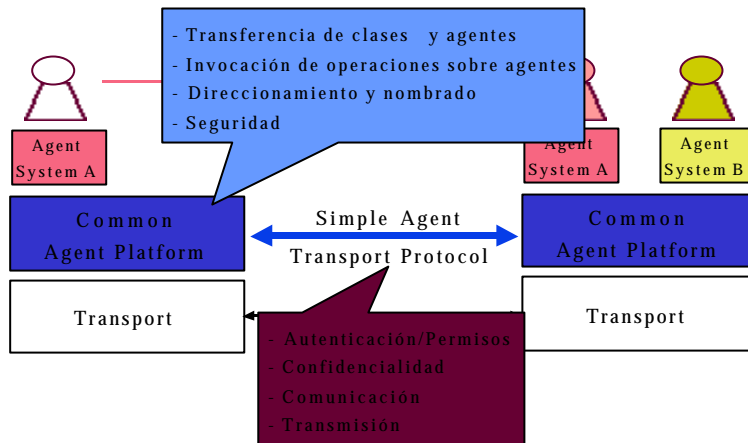
- Modelo más flexible basado en pares atributo-valor
- Permite ampliaciones para recoger nuevas necesidades

Agent Society

- Organización industrial fundada en 1996 (unos 50 socios)
- Objetivos
 - Facilitar la colaboración y transferencia de tecnología mediante el intercambio de información de desarrollo de agentes
 - Desarrollo de estándares de agentes abiertos e interoperabilidad
 - Gran colaboración con FIPA
- Áreas de trabajo
 - Arquitecturas de plataformas de agentes
 - Protocolos de transferencia de agentes
 - Lenguajes de programación de agentes
 - Lenguajes de comunicación de agentes

Agent Society

Arquitectura de plataforma común de agentes



© JPM, UCM 2001

Estandarización de Sistemas de Agentes

61

Agent Society

- El éxito de CAP y SATP dependerá de su compatibilidad con las especificaciones de MASIF y FIPA
 - Evolucionando hacia la interoperabilidad y estándares abiertos
- Información:

☞ <http://www.agent.org>

© JPM, UCM 2001

Estandarización de Sistemas de Agentes

62

Algunas conclusiones

- Todos los estándares tratan fundamentalmente los aspectos de comunicación entre agentes
- Actualmente FIPA parece el estándar más relevante
- Difíciles de implementar y costosos en eficiencia
- No hay una infraestructura (herramientas) estándar de servicios para facilidades como nombrado, registro, brokering o trading
 - FIPA intenta paliar este problema
 - Pero las distintas plataformas no son interoperables
- ¿Cómo se adaptan los ACL a las tecnologías de Internet?

Bibliografía

- T. Finin, Y. Labrou, J. Mayfield: *KQML as an agent communication language*, en Jeff Bradshaw (ed.), *Software Agents*, MIT Press, Cambridge (1995)
- T. Finin, Y. Labrou, *Agent Communication Languages*, tutorial en ASA/MA, 3 oct. 1999
- *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/papers/kqml93.pdf>
- *FIPA Abstract Architecture Specification*, <http://www.fipa.org>
- *FIPA Agent Management Specification*, <http://www.fipa.org>
- <http://www.cs.umbc.edu/acl/> Lenguajes de comunicación de agentes