

*Marco de trabajo y
entorno de desarrollo de Agentes*

*basado en la exposición de
Fabio Bellifemine*

Telecom Italia Lab - Torino (Italy)

ETAPS 2001, 7th Apr. 2001





Indice

- ★ FIPA – Foundation for Intelligent Physical Agents
- ★ JADE
 - Qué es
 - características principales, modelo arquitectónico, subsistemas
 - Qué viene después
- ★ Conclusiones



Foundation for Intelligent Physical Agents

FIPA

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

- producción de especificaciones para mover la tecnología de agentes
 - Agent Management and platform services
 - Agent Communication Model and Language
 - conjunto de “Interaction Protocols” comunes, (por ejemplo: Constraint Choice Language, ...)
- el núcleo de FIPA es el intercambio de mensajes que involucran
 - actos comunicativos,
 - lógica de predicados y
 - ontologías públicas,
 - ⇒ formas estándar de interpretar la comunicación entre agentes



FIPA

Modelo conceptual de una Plataforma de Agentes



Servicios normativos

Plataforma de Agentes

proporciona

Servicio

Servicios opcionales

Gestión de ciclo de vida

Servicio de páginas blancas

Servicio de páginas amarillas

Servicio de Transporte de mensajes (MTS)

Integracion de agentes software

Servicio de ontologia

Integracion hombre maquina





FIPA

modelo conceptual de comunicacion de agentes



EnvelopeEncodingScheme

Envelope

1

isTransmittedOver

1

TransportProtocol

1

contains

ACLEncodingScheme

1..*

Message

isExpressedIn

1

ACL

contains

CLEncodingScheme

Content

isExpressedIn

ContentLanguage

1

contains

InteractionProtocol

0..*

Symbol

belongsTo

Ontology

1..*

1

0..*

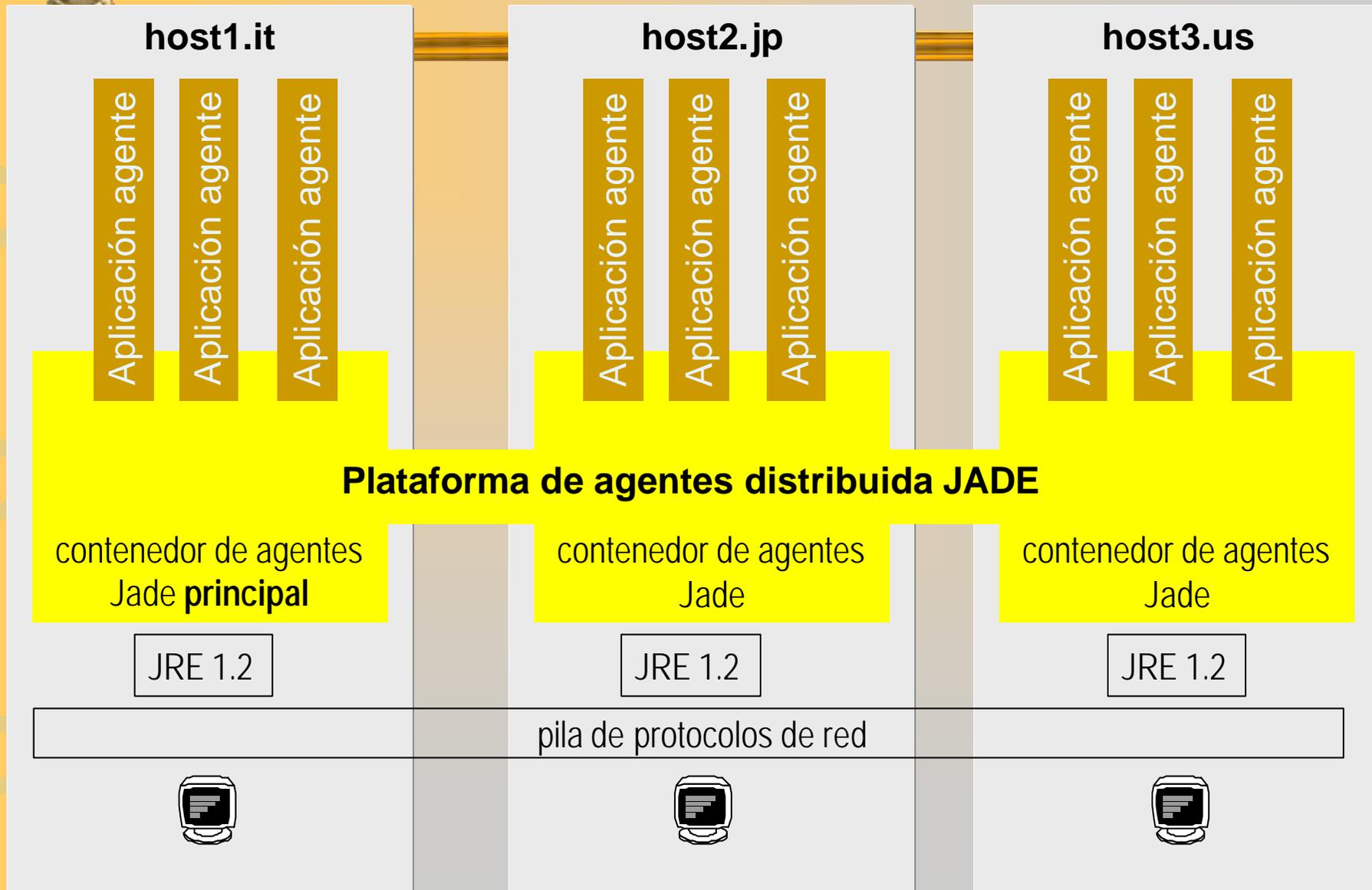


Middleware: JADE

- ★ usuarios: programadores de agentes MAS
- ★ servicios de agentes
 - ciclo de vida, paginas blancas, paginas amarillas, ...
- ★ herramientas para depuraci3n
 - agente monitor remoto, agente tonto, sniffer agente,
- ★ escalable
- ★ cumplimenta FIPA
- ★ implementado en Java
- ★ distribuido en codigo fuente bajo LPGL

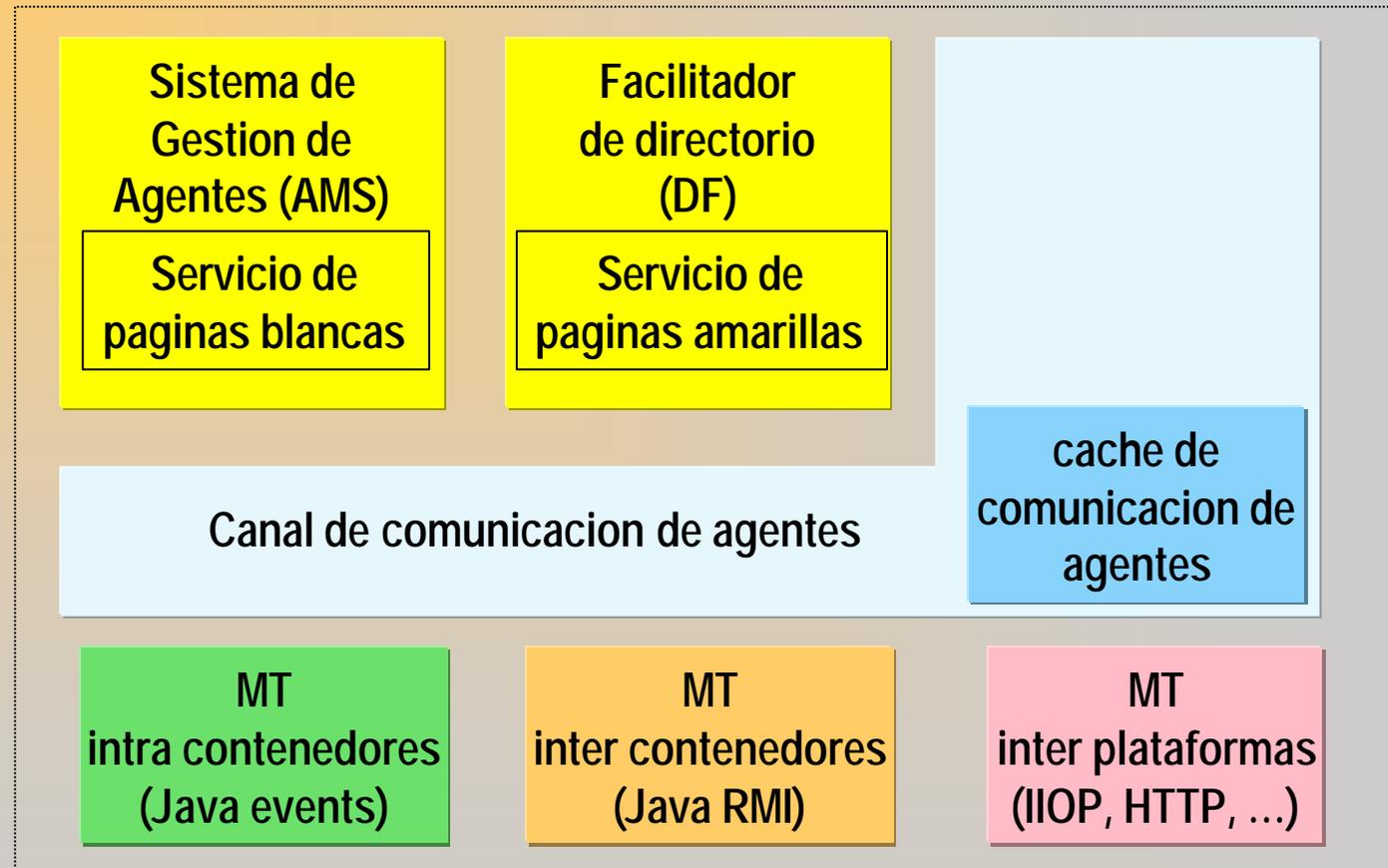


Arquitectura distribuida de la plataforma JADE





Arquitectura interna de la plataforma de ejecucion JADE



Nota: La arquitectura interna de un contenedor JADE es similar pero no contiene el AMS y el DF.



Concurrencia en JADE

- ★ varios contenedores por plataforma, pero 1 JVM por contenedor
- ★ varios agentes por contenedor con multihilo JVM
- ★ varios comportamientos en el mismo agente
 - planificados de modo no apropiativo
 - sin pila de retorno
 - que pueden componerse en un arbol
 - con cada comportamiento como un AF
 - un estado por slot de ejecucion de tiempo



JADE - Message Transport Service

- ★ controla la cola privada de mensajes ACL
- ★ el mecanismo de transporte se selecciona segun la situacion
 - para mejorar las prestaciones del paso de mensajes
- ★ canal de comunicacion de agentes distribuido
 - Se puede activar o desactivar cada protocolo de transporte de mensaje (Message Transport Protocols, MTP) en cualquier momento
 - IIOP basado en Sun ORB.
 - IIOP basado en ORBacus
 - permite persistencia de referencias a objeto.
 - permite una referencia URL maas natural
corbaloc: iiop: nombreHost: puerto/nombre
 - HTTP MTP de EPFL bajo LGPL
 - se han implementado varias codificaciones del ACL
 - textuales, basadas en XML (EPFL), compactas (Sonera)

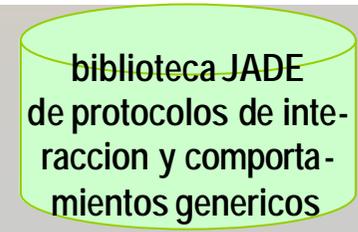
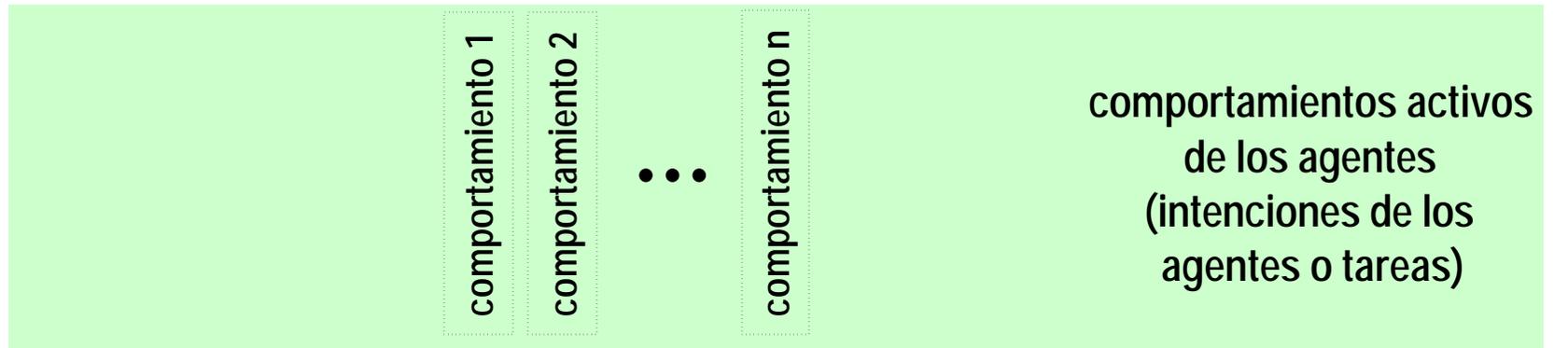


Modelo de Ejecucion de Agentes

implicaciones en la implementacion de JADE

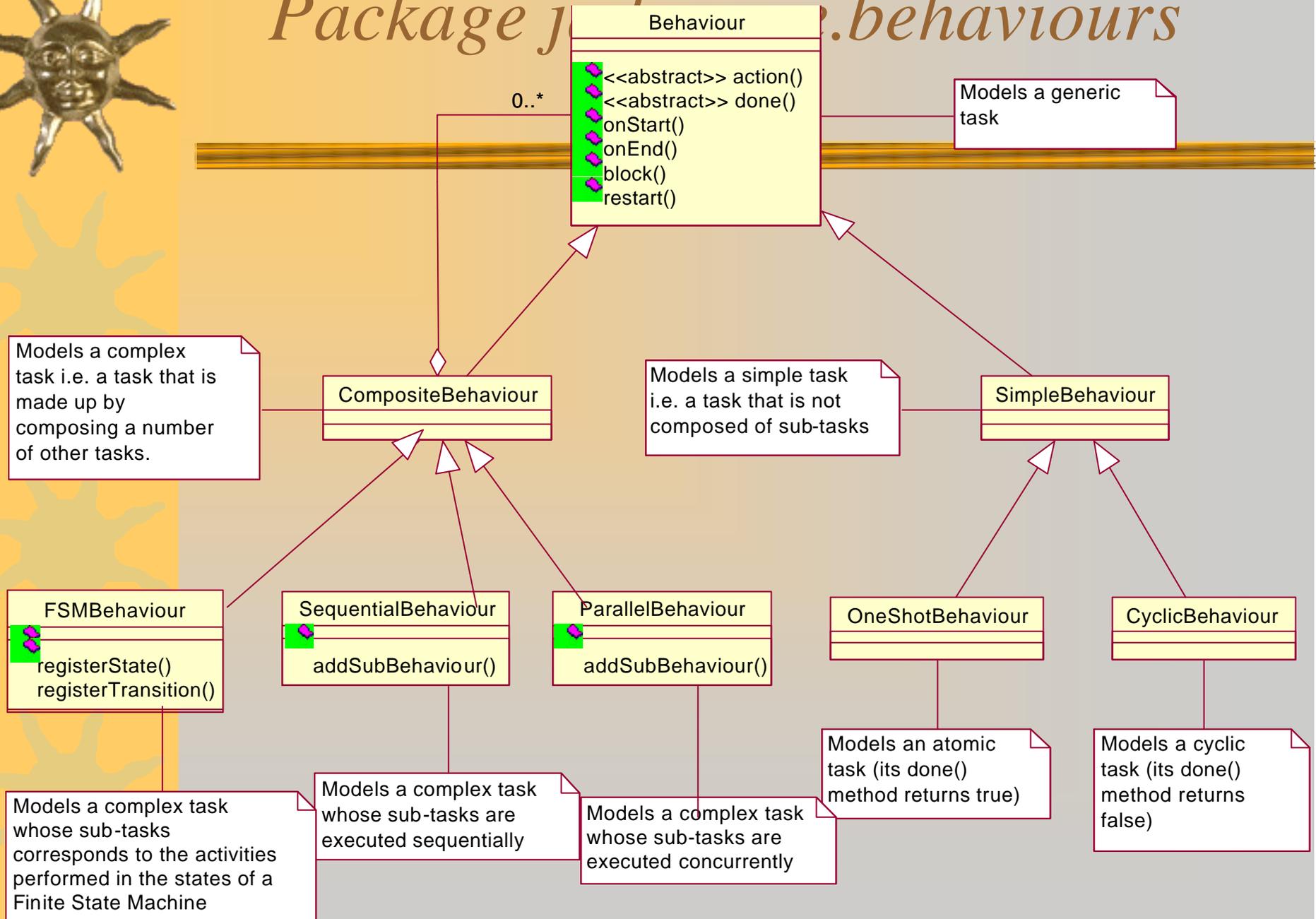
- ★ el agente es autonomo
 - controla su hilo de ejecucion
 - decide si leer, cuando y que mensajes leer
- ★ el agente precisa concurrencia
 - puede entablar varias conversaciones simultaneas
 - puede ejecutar varias tareas concurrentes
- Modelo de Programacion
 - Cada agente se correlaciona sobre una subclase Java de usuario, que hereda de Agent del paquete `jade.core`.
 - Cada tarea de agente se correlaciona sobre una subclase clase Java de Behaviour class en el paquete `jade.core.behaviours`.

Arqu. interna de un agente JADE





Package *java.lang*.behaviours





Modelo de comunicacion de agentes

- ★ Los agentes envian/reciben objetos java que representan mensajes ACL, dentro del alcance de los protocolos de interaccion.
 - JADE oculta el interior del codec
 - Nivel Envelope : textual, basado en XML
 - Nivel ACL: textual, basado en XML, compacto
 - Nivel de Content Language:
 - FIPA SL-0 + API para registrar otros lenguajes de contenido.
 - Soporta serializacion directa Java Base64
 - Nivel Ontology
 - Gestion de Agentes FIPA; Gestion de Agentes JADE
 - API para registrar lenguajes de contenido propios.
 - JADE provee una biblioteca de protocolos de interaccion habituales
 - El usuario solo tiene que implementar los manejadores
 - y puede componer las tareas del agentes como super-estados de un Automata Finito



Movilidad de agentes

- ★ JADE soporta movilidad y clonacion intra-plataforma
 - que involucra varios hosts
 - cada uno con su contenedor
 - a peticion del propio agente
 - doMove(ubicacion) / doClone(ubicacion, String)
 - before/afterMove() before/afterClone()
 - a traves de AMS a la plataforma
 - segun el protocolo de interaccion
 - jade.domain.MobilityOntology define todos los conceptos y acciones precisas para soportar movilidad de agentes y clonacion



Herramientas graficas de apoyo al desarrollo

- ★ RMA (Remote Monitoring Agent)
 - consulta paginas blancas locales y de otros contenedores remotos
 - control del ciclo de vida del agente (ej. creacion remota, migracion, ...)
 - activacion y desactivacion de los MTP en los contenedores
- ★ GUI DF (Directory Facilitator)
 - consulta de paginas amarillasto browse the yellow-page service
 - para federarse y consultar otros DF remotos
- ★ Agente tonto (envia/recibe guarda/exporta ACLMessages)
- ★ Agente fisgon (sniffer)
 - fisga, depura, almacena conversaciones multi-agente
- ★ Agente Introspector
 - depura la cola de mensajes de E/S, la cola de comportamientos, ...



herramientas de apoyo grafico

The screenshot displays the JADE Remote Agent Management GUI with several overlapping windows:

- DF df**: A window with a "Commands" menu and tabs for "General", "Current message", and "Queued message". It includes buttons for "Reset", "Send", "Open", "Save", "Open queue", "Save queue", "Set as current", "Reply", "View", and "Delete".
- Current message**: Shows message details for "da0".
 - Sender: da0
 - Receiver: [empty]
 - Communicativ...: accept-proposal
 - Content: [empty text area]
 - Language: [empty]
 - Ontology: [empty]
 - Protocol: [empty]
 - Conversation: [empty]
 - In-reply-to: [empty]
 - Reply-with: [empty]
 - Reply-by: [empty]
 - Envelope: [empty]
- JADE Remote Agent Management GUI**: The main interface showing a tree view of agents:
 - JADE
 - Container-1
 - rma@ior:000000000000
 - Front-End
 - df@ior:000000000000
 - acc@ior:000000000000
 - ams@ior:000000000000
 - Agostino@IOR:00000000
 - Giovanni@IOR:00000000

- VIEW REGISTERED AGENT DESCRIPTIONS**: A window titled "DF AGENT DESC...DESCRIPTION" showing details for the "Giovanni" agent:
- agent-name: Giovanni@IOR:0000000
- agent-address: IOR:0000000000000016
- df-state: active
- ontology: pa-ontology
- agent-type: fipa-agent
- ownership: Giovanni
- language: [empty]
- interaction-protoc...: fipa-request fipa-Contrac
- AGENT SERVICES:
 - service-name: AppointmentScheduling
 - service-ontology: pa-ontology
 - negotiable-propert...: [empty]
 - Service no. 1 of 1
 - service-type: personal-agent
 - fixed-properties: (represents Giovanni)
 - communication-pr...: [empty]
- Agostino - Appointment Scheduler**: A window with a "Directory Appointment" section and a calendar for the month of September 1999.

| S | M | T | W | T | F | S |
|----|----|----|----|----|----|----|
| | | | | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | |



Escalabilidad en JADE

- ★ Configuración de una plataforma
 - de un MAS en un solo host
 - a un agente en un solo host (para un cluster)
 - ✓ se puede cambiar en ejecución: en caliente gracias a las caches locales (por nombre)
- El contenedor principal no es un cuello de botella
 - dado que el canal de comunicación de agentes es distribuido
 - y el contenedor principal solo se usa cuando es necesario



Aspectos de FIPA enmascarados al usuario

- ★ Plataforma de agentes ya construida (AMS, DF, y ACC)
- ★ Ontología y servicios de gestion de agentes
 - la plataforma registra el agente, dado un nombre y direccion
 - la clase Agent proporciona una interfaz simple a los servicios del DF (check-in, busqueda, ...)
- ★ MTP y parsers ACL y SL ya implementados
- ★ No es preciso implementar Protocolos de Interaccion
 - solo es preciso aumentarlos.