

# JADE – Directory Facilitator

## Taller de sistemas multiagentes

Prof. Dr. Ariel Monteserin  
amontese@exa.unicen.edu.ar  
ISISTAN –Fac. Cs. Exactas – UNICEN  
Tandil, Argentina



1

## Limitaciones observadas

- ¿Qué limitaciones encuentran al intentar comunicarse con otro agente?

Debemos especificar el nombre del agente con el cual nos queremos comunicar dentro del comportamiento del agente

```
msg.addReceiver (new AID ("X1",  
AID.ISLOCALNAME) );
```



2

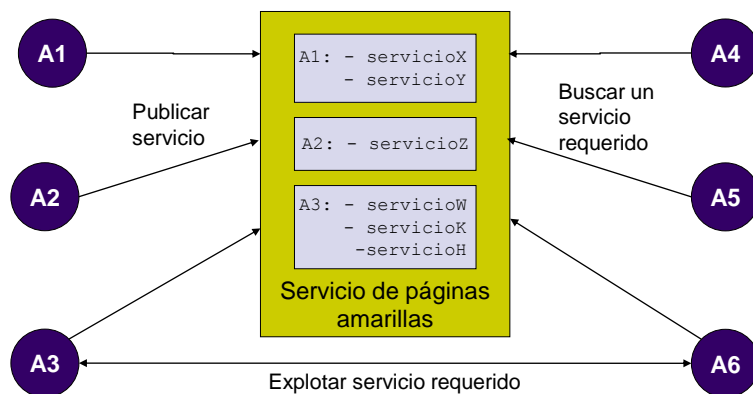
## Agente DF



- Provee el servicio de *Paginas Amarillas*
  - Permite a los agentes publicar y buscar servicios.
- Existe un único DF por plataforma JADE.
- Interacción mediante mensajes ACL.
  - Lenguaje: SL0
  - Ontología: FIPA-agent-management

3

## Servicio de páginas amarillas



4

## Publicar un servicio



- Para publicar un servicio el agente debe proveer al DF:
  - AID
  - Lenguaje y ontología para interactuar con el agente
  - Para cada servicio:
    - Nombre
    - Lenguaje y ontología para explotarlo
    - Propiedades del servicio



5

## Clases relacionadas al DF



- `DFAgentDescription`
- `ServiceDescription`
- `DFService`

6

## DFAgentDescription



- Representa la descripción de un agente en el catálogo DF
  - setName(AID a)
  - addServices(ServiceDescription s)
  - addLanguages(String s)
  - addOntologies(String s)
  - addProtocols(String s)

```
Al: - servicioX  
    - servicioY
```

7

## ServiceDescription



- Modela un servicio de un agente
  - addLanguages(String s)
  - addOntologies(String s)
  - addProtocols(String s)
  - addProperties(Property p)
    - match(Property p)
    - setName(String s), setValue(Object o)
  - setName(String s)
  - setType(String s)

8

## DFService



- Provee un set de métodos estáticos para comunicarse con los servicios del DF

- `register(Agent a, DFAgentDescription d)`
  - Registra una nueva descripción en el DF.
  - `FIPAException`
- `deregister(Agent a)`
- `search(Agent a, DFAgentDescription d)`
  - Devuelve un array de `DFAgentDescription`.
  - `d` es utilizado con un template para la búsqueda.
  - Además
    - `SearchConstraints` restringe el número de resultados.
    - Puede setearse un timeout de espera.



9

## Publicar un servicio



```
protected void setup() {
    ...
    // Registra el servicio en las paginas amarillas
    DFAgentDescription dfd = new DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new ServiceDescription();
    sd.setType("negociacion");
    sd.setName("peliculas");
    dfd.addServices(sd);
    try {
        DFService.register(this, dfd);
    }
    catch (FIPAException fe) {
        fe.printStackTrace();
    }
    ...
}
```

10

## Quitar un servicio



- Deben ser eliminados cuando el agente es dado de baja:

```
protected void takeDown() {  
    // Quita el servicio del DF  
    try {  
        DFService.deregister(this);  
    }  
    catch (FIPAException fe) {  
        fe.printStackTrace();  
    }  
}
```

11

## Publicación de servicios



**Incorporar al agente RESPONDER  
la publicación del servicio  
en el DF (para el FINAL)**



- Descripción del servicio:
  - Nombre: "películas"
  - Tipo: "negociacion"

12

## Buscar servicios



- Un agente debe utilizar una descripción template
  - Resultado: lista de todas las descripciones que concuerdan con el template.

```
DFAgentDescription template = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType("negociacion");
Sd.setName("peliculas");
template.addServices(sd);
try {
    DFAgentDescription[] result = DFService.search(myAgent,
        template);
    ...
}
catch (FIPAException fe) { fe.printStackTrace(); }
```



13

## Búsqueda de servicios



Incorporar al agente INITIATOR  
la búsqueda del servicio  
en el DF (para el FINAL)



- Descripción del servicio:
  - Nombre: "peliculas"
  - Tipo: "negociacion"

14

## DF Subscription Services



- Un agente puede solicitar al DF un aviso cuando un determinado servicio sea registrado:

```
DFAgentDescription dfd = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType(...);
dfd.addServices(sd);
SearchConstraints sc = new SearchConstraints();
sc.setMaxResults(new Long(1));
send(DFService.createSubscriptionMessage(this,
    getDefaultDF(), dfd, sc));
```

15

## DF Subscription Services



```
DFAgentDescription template = // template de busqueda
addBehaviour( new SubscriptionInitiator( this,
    DFService.createSubscriptionMessage( this, getDefaultDF(),
        template, null)
    {
        protected void handleInform(ACLMessage inform) {
            try {
                DFAgentDescription[] dfds =
                    DFService.decodeNotification(inform.getContent());

                // procesar dfds...
            }
            catch (FIPAException fe) {fe.printStackTrace(); }
        }
    }
));
```

Permitir que el agente INITIATOR se suscriba para recibir alertas cuando un nuevo RESPONDER se registre.



17



## Cancelar suscripción



- DFService.

```
createCancelMessage (Agent a,  
                    AID dfName,  
                    ACLMessage subscribe)
```

- dfName – AID del DF - getDefaultDF()
- subscribe – El mensaje de suscripción previamente enviado al DF.