

Contenidos

- ⇒ Importancia de la comunicación entre agentes
- ⇒ Modelos de comunicación
- ⇒ Teoría de los actos del habla
- ⇒ Organizaciones de estandarización
 - × OMG, KSE y FIPA
- ⇒ Lenguajes de comunicación de agentes (ACLs)
 - × KQML y FIPA-ACL
- ⇒ Protocolos de Interacción
- ⇒ Desarrollo de Ontologías

Comunicación entre Agentes

- ⇒ **La comunicación entre agentes es la llave para obtener todo el potencial del paradigma de agentes**
 - ✖ Al igual que el desarrollo del lenguaje humano fue la llave para el desarrollo de la inteligencia y de la sociedad
- ⇒ Los agentes emplean un lenguaje de comunicación (ACL - Agent Communication Language) para comunicar información y conocimiento
- ⇒ Las sociedades de agentes pueden realizar tareas que los agentes individuales no pueden
- ⇒ Los sistemas multiagente (MAS) están orientados a la resolución distribuida de problemas
 - ✖ Sólo es posible si los agentes tienen la capacidad de comunicación sobre la que establecer estrategias de cooperación

Modelos de Comunicación (1)

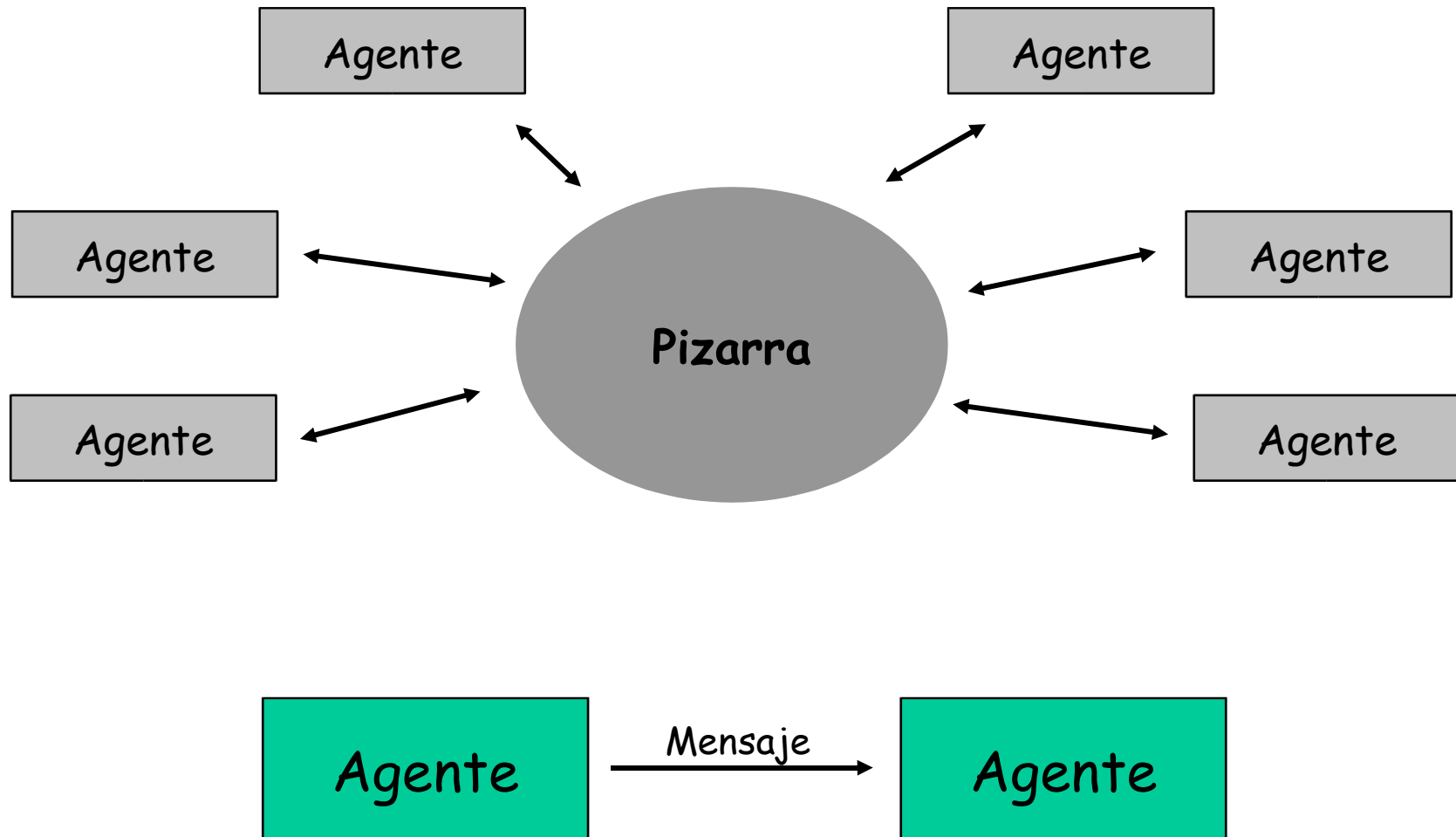
⇒ Arquitecturas de Pizarra

- ✗ Zona de trabajo común en la que se encuentra la información a compartir
- ✗ No hay comunicación directa entre los agentes
- ✗ Pueden existir agentes con tareas de control específicas
- ✗ Pueden existir varias pizarras
- ✗ Inconveniente: comunicación centralizada

⇒ Paso de Mensajes

- ✗ La comunicación se establece directamente entre dos agentes (emisor y receptor)
- ✗ Pueden existir agentes facilitadores
- ✗ Ventaja: flexibilidad

Modelos de Comunicación (2)



Comunicación tipo Vulcano (1)

- ⇒ Vulcano es el planeta de Spock (Star Treck)
- ⇒ Esta comunicación consiste en la transmisión de pensamientos o conocimiento de una mente a otra
 - ✘ "Cuando conectas tu cerebro al mío tu sabes lo que yo sé"
- ⇒ Utilizada en la comunicación máquina-máquina
 - ✘ Montar nuevos discos en una máquina
 - ✘ Formularios EDI permiten a un programa rellenar datos en las variables de otro programa
 - ✘ Las BD distribuidas

Comunicación tipo Vulcano (2)

⇒ No sirve para modelar la comunicación humana por:

1. Mis pensamientos no son los tuyos

- ✗ No te servirán de nada a menos que se expresen en un lenguaje común y todo lo que se asuma sea conocimiento compartido

2. Se ignora la naturaleza volitiva de los actos comunicativos

- ✗ En cada momento particular seleccionaré las cosas que "deseo" comunicar.

3. No tengo por qué comunicarte mis pensamientos

- ✗ Tengo que decirte lo que quiero que tu hagas con las proposiciones que yo expreso. Si no eres capaz de diferenciar entre conjeturas, informaciones o preguntas, mis expresiones no te servirán de nada.
- ✗ "Una comunicación efectiva necesita un correcto reconocimiento de los actos del habla"

Teoría de los Actos del Habla (1)

- ⇒ Está desarrollada por lingüistas para ayudar en la comprensión del lenguaje humano
- ⇒ Estudia el lenguaje como acción
- ⇒ Los hablantes no sólo expresan sentencias ciertas o falsas
- ⇒ Los hablantes realizan actos del habla
 - × Informan, preguntan, sugieren, prometen...
- ⇒ Cada expresión conlleva un acto del habla
- ⇒ Identificar el acto del habla es imprescindible para una correcta comunicación

Teoría de los Actos del Habla (2)

- ⇒ Es la base teórica en la comunicación entre agentes
- ⇒ En un acto del habla se encuentran 3 acciones:
 - × **Locución.** Contexto del hecho físico del acto del habla
 - × **Ilocución.** Intención con la que se realiza
 - × **Perlocución.** Acto que ocurre como resultado
- ⇒ Ejemplo: "Cierra la ventana"
 - × Locución: identificación del emisor, del receptor, de la ventana...
 - × Ilocución: el emisor quiere que el receptor cierre la ventana
 - × Perlocución: el receptor cierra (o no) la ventana

La hipótesis F(P)

⇒ Todo acto del habla consiste en una fuerza F aplicada a una proposición P

Ilocución	Sentencia
Información	La ventana está abierta
Pregunta	¿Está abierta la ventana?
Orden	Abre la ventana
Petición	Podrías abrir la ventana
Promesa	Te prometo que abriré la ventana
Oferta	Abriré la ventana si tu quieres
...	...

Organizaciones de Estandarización

⇒ OMG (Object Management Group)

× <http://www.omg.org>

⇒ KSE (Knowledge Sharing Effort)

× <http://www.cs.umbc.edu/kse/>

⇒ FIPA (Foundation for Intelligent and Physical Agents)

× <http://www.fipa.org>

OMG

⇒ Asociación de empresas e instituciones formada a finales de los 80

⇒ Objetivos

- ✘ Reutilización, portabilidad e interoperabilidad de sistemas distribuidos de componentes software orientados a objetos

⇒ Resultados

- ✘ UML (*Unified Modeling Language*)
- ✘ CORBA (*Common Object Request Broker Architecture*)
- ✘ MASIF (*Mobile Agent System Interoperability Facility*)

KSE (1)

⇒ Iniciativa

- ✗ ARPA (*Advanced Research Projects Agency*)
- ✗ ASOFR (*Air Force Office of Scientific Research*)
- ✗ NRI (*Corporation for National Research Initiative*)
- ✗ NSF (*National Science Foundation*)

⇒ Objetivo

- ✗ Facilitar la compartición y reutilización de bases y sistemas basados en conocimiento

⇒ Filosofía

- ✗ La interacción eficaz de agentes software tres componentes fundamentales
 1. Un lenguaje común
 2. Una comprensión común del conocimiento intercambiado
 3. Una habilidad para intercambiar todo lo relativo a (1) y a (2)

KSE (2)

⇒ El KSE está organizado entorno a tres grupos de trabajo

× *Interlingua Group*

☒ Encargado de desarrollar un lenguaje común para poder representar los contenidos de las bases de conocimiento

× *SRKB Group (Shared Reusable Knowledge Base)*

☒ Encargado de definir los conocimientos reutilizables para distintos dominios de aplicación

× *External Interfaces Group*

☒ Encargado de la interacción (en tiempo de ejecución) entre sistemas basados en conocimiento

⇒ Resultados

× Sintaxis: KIF (*Knowledge Interchange Format*)

☒ Lenguaje para el intercambio de conocimiento <http://www.cs.umbc.edu/kse/kif/>

× Semántica: Ontolingua

☒ Lenguaje para la definición de Ontologías <http://www.cs.umbc.edu/kse/ontology/>

× Pragmática: KQML (*Knowledge Query and Manipulation Language*)

☒ Lenguaje para la comunicación entre agentes <http://www.cs.umbc.edu/kqml>

☒ Interoperabilidad entre agentes en un entorno distribuido

KIF

- ⇒ Permite la representación de conocimiento
- ⇒ Basado en el cálculo de predicados de primer orden
- ⇒ Sintaxis similar a la empleada en LISP
- ⇒ Ejemplos de expresiones
 - × Datos
 - ⊗(salario 27843834k analista 30.000)
 - ⊗(salario 122333212s operador 15.000)
 - × Conocimiento
 - ⊗(> (salario empleado1) (salario empleado2))
 - ⊗(=> (and (num-real ?x) (num-par ?n)) (>(expt ?x ?n) 0))
 - ⊗(defrelation soltero(?x) := (and(hombre ?x) (not(casado ?x))))
 - × Inferencia
 - ⊗(interested Agente1 `(salario ,?x ,?y ,?z))

Ontolingua

- ⇒ Lenguaje para construir, publicar y compartir ontologías
- ⇒ Las ontologías pueden traducirse automáticamente a otros lenguajes de contenido (KIF, Prolog, CLIPS...)
- ⇒ Incluye primitivas para combinar ontologías
- ⇒ Este grupo ha definido un número considerable de ontologías sobre distintos dominios que pueden ser de interés para distintas aplicaciones

KQML

⇒ En la comunicación intervienen los siguientes elementos

- × El protocolo de interacción

- ☒ Estrategia de alto nivel seguida por el agente software para controlar la interacción con otros agentes
- ☒ Desde esquemas de negociación hasta esquemas tan simples como: "cada vez que desconozcas algo de tu interés busca a algún agente que lo sepa y pregúntale"

- × El lenguaje de comunicación

- ☒ Es el medio a través del que se intercambian los actos de la comunicación
- ☒ Indica si el contenido de la comunicación es una información, una respuesta o algún tipo de consulta

- × El protocolo de transporte

- ☒ Mecanismo de transporte utilizado en la comunicación
- ☒ TCP, SMTP, HTTP...

KQML (capas)

⇒ El lenguaje KQML está dividido en 3 capas

- × La capa de contenido

- ☒ Relacionada con el contenido del mensaje
- ☒ Puede expresarse en cualquier lenguaje
- ☒ KQML sólo tiene interés en identificar el inicio y el final del contenido

- × La capa de mensaje

- ☒ Es el núcleo del lenguaje KQML
- ☒ Determina los tipos de interacción que puede realizar un agente
- ☒ Identifica el protocolo, la ontología y el acto del habla (performative)

- × La capa de comunicación

- ☒ Identifica las características del mensaje que describen los parámetros de bajo nivel de la comunicación (emisor, receptor e identificador de mensaje)

KQML (mensajes)

- ⇒ Un mensaje en KQML representa la intención de realizar alguna acción (performative o acto del habla)
- ⇒ Un mensaje KQML es una lista tipo LISP en la que
 - ✗ El primer elemento de la lista es el identificador del acto del habla
 - ✗ El resto de la lista son pares atributo-valor sin un orden predefinido

```
(ask-one
  :sender      agente1
  :receiver    servidor-telefonica
  :reply-with  accion-telefonica
  :content     (Precio Telefonica ?p)
  :language    prolog
  :ontology    IBEX-35
)
```

KQML (atributos reservados)

Atributo	Significado
:content	La información
:sender	Emisor del mensaje
:receiver	Receptor del mensaje
:language	El nombre del lenguaje de representación empleado en el atributo :content
:ontology	El nombre de la ontología utilizada en el atributo :content
:reply-with	Etiqueta para la respuesta (si es que el emisor la espera)
:in-reply-to	La etiqueta esperada en la respuesta

KQML (performatives)

Categoría	Nombre
Información	tell, untell
Consulta	evaluate, reply, ask-if, ask-about, ask-one, ask-all, sorry
Consulta con respuesta múltiple	stream-about, stream-all
Orden	achieve, unachieve
Control de respuestas	standby, ready, next, rest, discard
Capacidad	advertise
Notificación	subscribe, monitor
Gestión de red	register, unregister, forward, broadcast, pipe, break
Facilitación	broker-one, broker-all, recommend-one, recommend-all, recruit-one, recruit-all

KQML (performatives)

Nombre	Significado
tell	E comunica una información que se encuentra en su BC
evaluate	E quiere que R evalúe la expresión
reply	E comunica a R una respuesta esperada
ask-if	E quiere saber si la expresión se encuentra en la BC de R
ask-about	E quiere conocer todo el conocimiento de R relacionado con la expresión
ask-one	E quiere que R conteste a una pregunta
stream-about	Versión en respuesta múltiple de ask-about
achieve	S quiere que R convierta en verdadera la expresión en su BC
standby	S quiere que R esté preparado para responder a una <i>performative</i>
ready	S está preparado para responder a R
next	S quiere que R le devuelva la siguiente respuesta (de una consulta múltiple)
advertise	S está preparado para responder a una determinada <i>performative</i>
register	S puede responder a <i>performatives</i> de un determinado agente
broadcast	S quiere que R envíe la <i>performative</i> a todas los agentes conectados
recommend-one	S quiere el nombre de un agente que pueda contestar a una <i>performative</i>
recruit-one	S quiere que R le proporcione el nombre de otro agente que conteste a la <i>preformative</i>

KQML (ejemplo 1)

A envía a B:

(ask-one :content (interested '(sobre, bloqueA, ?x)) :
language KIF :ontology bloques :reply-with q1)

B contesta a A:

(reply :content (sobre bloqueA bloqueB) :language KIF :
ontology bloques :in-reply-to q1)

KQML (ejemplo 2)

A envía a B:

(stream-about :content bloqueA :language KIF :
ontology bloques :reply-with q1)

B contesta a A con la siguiente serie de performatives:

(tell :content (sobre bloqueA bloqueB) :language KIF :
ontology bloques :in-reply-to q1)

(tell :content (sobre mesa bloqueA) :language KIF :
ontology bloques :in-reply-to q1)

(sorry :in-reply-to q1)

KQML (ejemplo 3)

A envía a B:

(achieve :content (sobre mesa bloqueB) :language KIF
:ontology bloques :reply-with q1)

B después de llevar a cabo la tarea contesta a A:

(tell :language KIF :ontology bloques :in-reply-to q1 :
content (sobre mesa bloqueB))

KQML (ejemplo 4)

A envía a B:

(standby :language KQML :ontology K10 :reply-with q1
:content (stream-about :language KIF :ontology bloques :
reply-with q3 :content bloqueA))

B responde con:

(ready :reply-with XA77 :in-reply-to q1)

A continúa con:

(next :in-reply-to XA77)

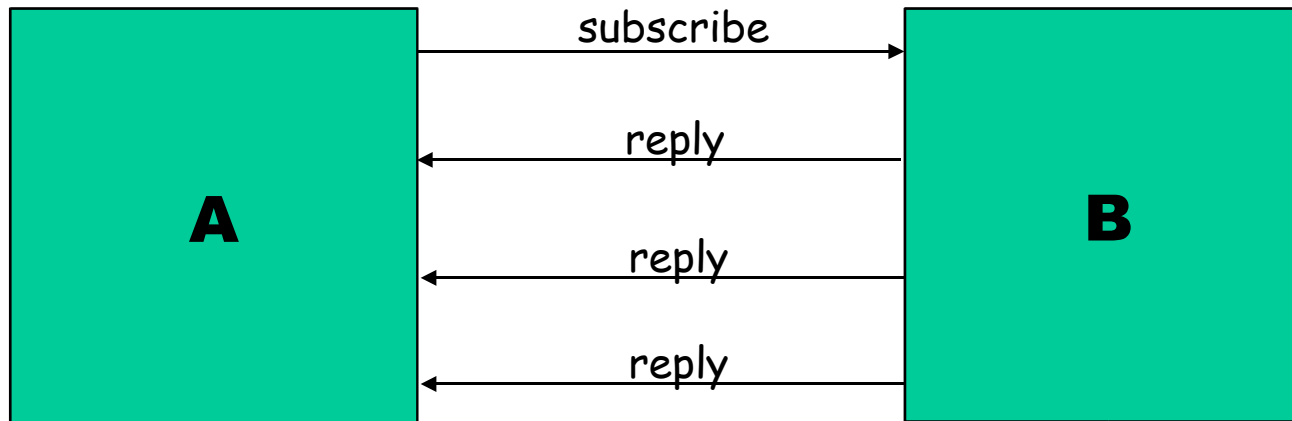
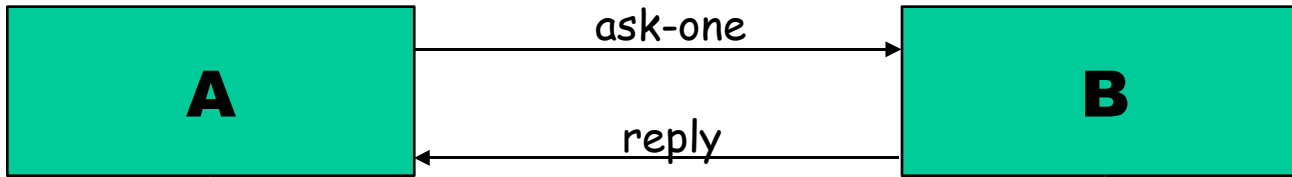
B responde con:

(tell :language KIF :ontology bloques :in-reply-to q3 :
content (sobre bloqueA bloqueB))

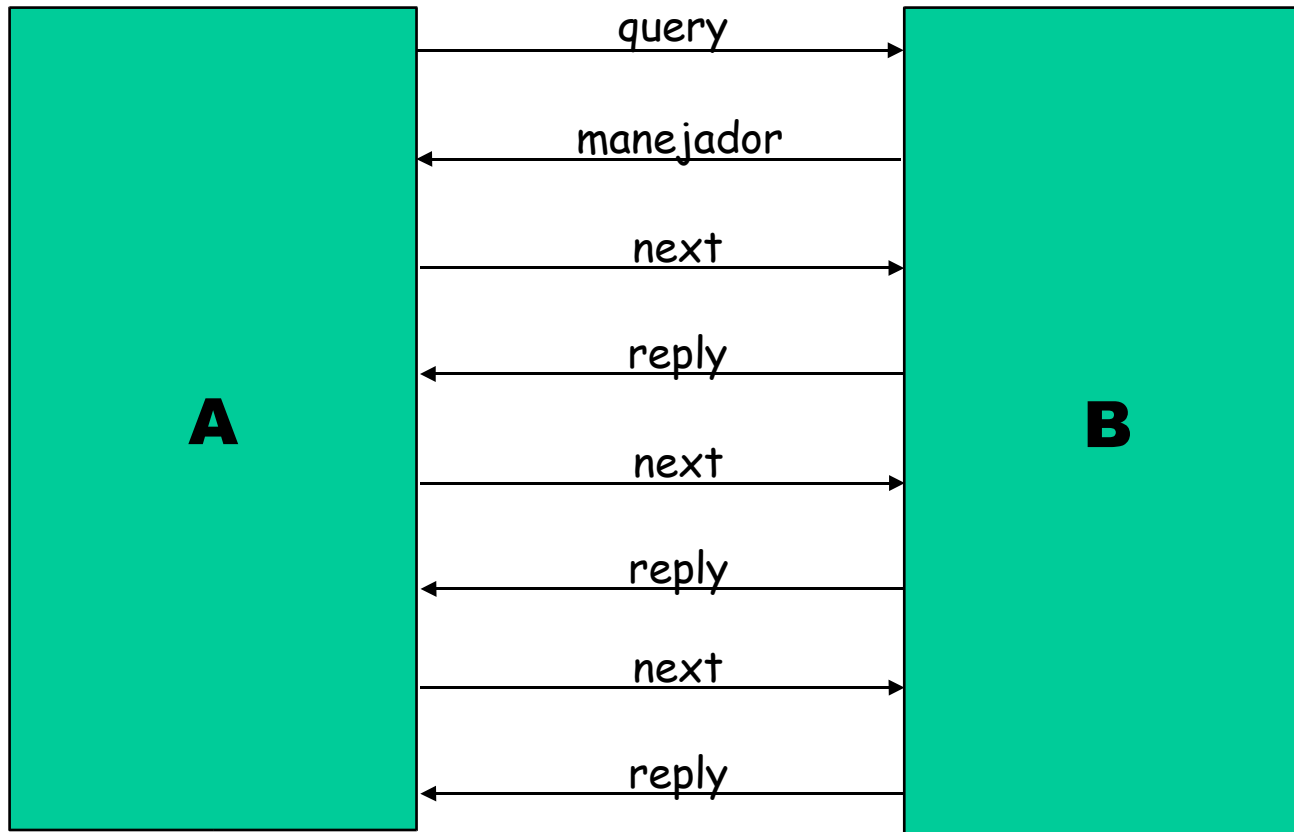
Así hasta que A envía:

(discard :in-reply-to XA77)

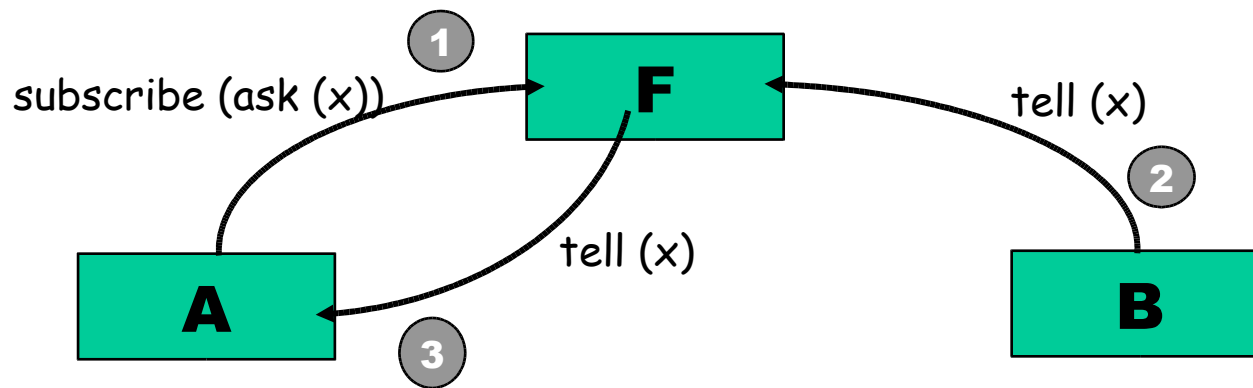
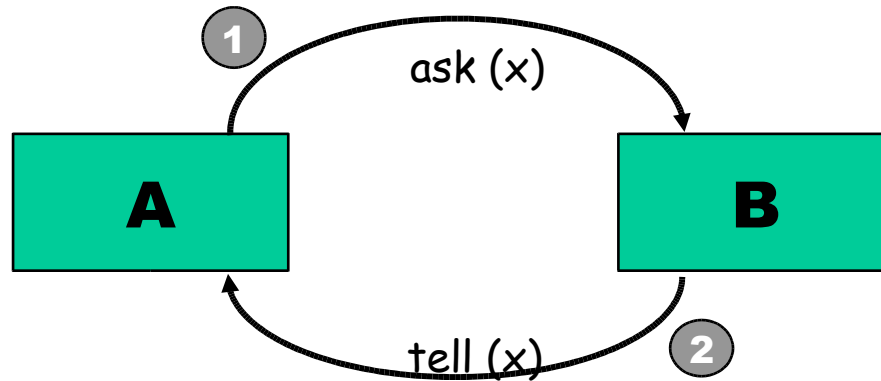
KQML (protocolos básicos)



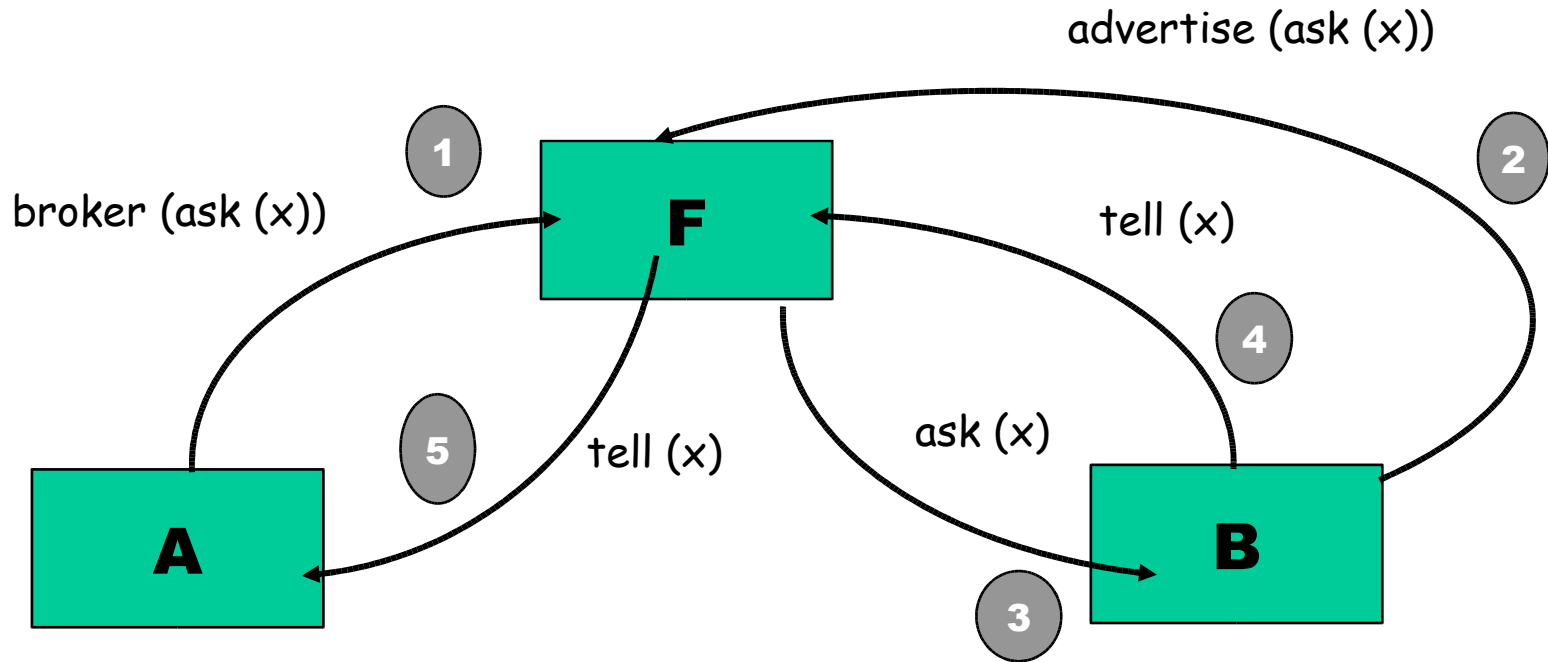
KQML (protocolos básicos)



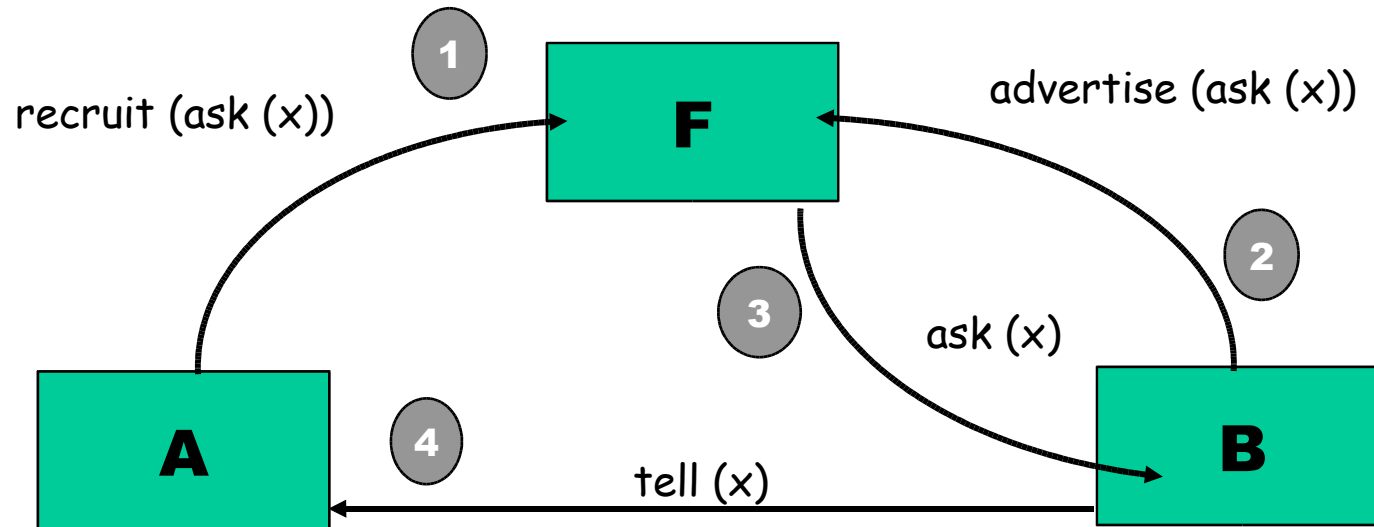
KQML (agentes facilitadores)



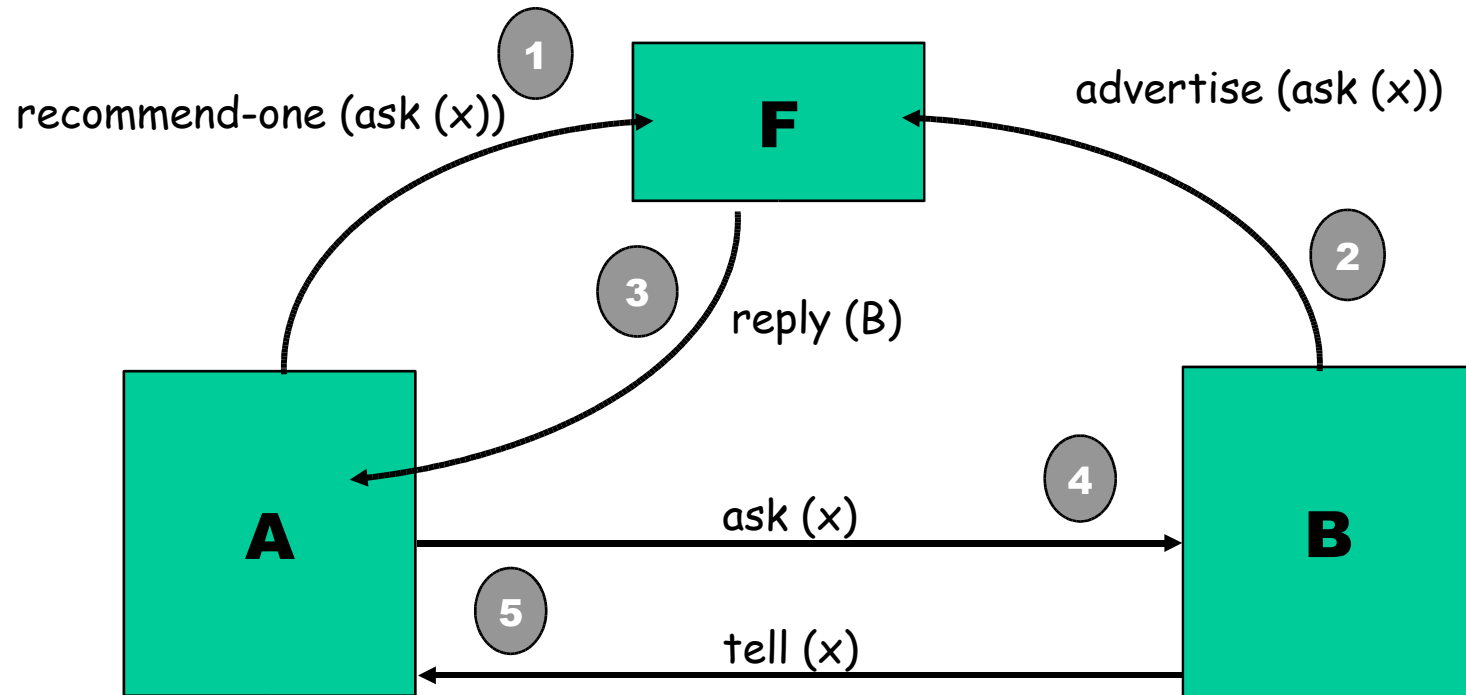
KQML (agentes facilitadores)



KQML (agentes facilitadores)



KQML (agentes facilitadores)



FIPA

⇒ Foundation for Intelligent and Physical Agents

- ✗ Organización internacional dedicada a la promoción de la industria de los agentes inteligentes mediante el desarrollo de especificaciones que soporten la interoperabilidad entre agentes y aplicaciones basadas en agentes

⇒ Cubre todos los aspectos de un entorno de agentes

- ✗ Aplicaciones
- ✗ Arquitectura
- ✗ Comunicación
 - ☒ Protocolos de Interacción, Actos comunicativos y Lenguajes de Contenidos
- ✗ Gestión de Agentes
- ✗ Transporte de mensajes

⇒ Hasta la fecha ha publicado 3 conjuntos de especificaciones

- ✗ FIPA 97, FIPA 98 y FIPA 2000

FIPA ACL

- ⇒ Basado en la Teoría de los Actos del Habla
- ⇒ Un mensaje en FIPA ACL representa la intención de realizar alguna acción (acto comunicativo)
- ⇒ Un mensaje en FIPA ACL tiene una sintaxis similar a un mensaje en KQML
 - ✗ El primer elemento de la lista es el identificador del acto comunicativo (obligatorio)
 - ✗ El resto de la lista son pares parámetro-valor sin un orden predefinido (opcionales)

```
(inform
  :sender          gestor-bolsa
  :receiver        agente1
  :in-reply-to    accion-telefonica
  :content         (Precio Telefonica 20)
  :language       prolog
  :ontology        IBEX-35
)
```

FIPA ACL (parámetros)

Atributo	Significado
:content	Contenido del mensaje
:sender	Identidad del Emisor del mensaje
:receiver	Identidad del Receptor del mensaje (un agente o una lista de agentes)
:language	El nombre del lenguaje de representación empleado en el atributo :content
:ontology	El nombre de la ontología utilizada en el atributo :content
:reply-with	Etiqueta para la respuesta (si es que el emisor la espera)
:in-reply-to	La etiqueta esperada en la respuesta
:protocol	Identificador del protocolo de interacción que se está utilizando
:conversation-id	Identificador de una secuencia de actos comunicativos que forman parte de una misma conversación
:reply-to	Agente al que han de ser enviadas las respuestas (si no es el emisor)
:reply-by	Indicación del tiempo en el que se quiere que se responda al mensaje

FIPA ACL (actos comunicativos)

Nombre	Paso de Información	Solicitud de Información	Negociación	Ejecución de Acciones	Manejo de Errores
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					

Protocolos de Interacción (1)

- ⇒ Para poder establecer una conversación entre agentes es necesario definir previamente el protocolo que van a seguir durante la conversación
- ⇒ Un protocolo de interacción es una descripción detallada del tipo y orden de los mensajes involucrados en una conversación entre agentes
- ⇒ Un agente puede participar simultáneamente en múltiples diálogos con diferentes agentes y con diferentes protocolos de interacción
- ⇒ Dentro del estándar FIPA (utilizando AUML) se encuentran definidos algunos de los protocolos más extendidos

Protocolos de Interacción (2)

- ⇒ Para utilizar un protocolo en una conversación, los agentes deben escribir el nombre del protocolo a utilizar en el parámetro : **protocol**
- ⇒ Un protocolo termina cuando
 - ✗ Se alcanza el último mensaje del protocolo o
 - ✗ Se elimina el nombre del protocolo del parámetro :protocol
- ⇒ Cuando un agente no conoce un determinado protocolo tiene que devolver un mensaje refuse explicando el motivo por el que rechaza la comunicación
- ⇒ Si durante el seguimiento de un protocolo, un agente recibe un mensaje no contemplado en el mismo tiene que devolver un mensaje not-understood
 - ✗ Está prohibido responder un *not-understood* con otro *not-understood* para no caer en bucles infinitos

AUML

⇒ AUML = Agent Unified Modelling Language

⇒ UML es insuficiente para modelar sistemas multiagente

- ✘ Comparados con los objetos, los agentes son activos ya que actúan por razones que emergen de ellos mismos
- ✘ Entre otros aspectos, necesitamos modelar los protocolos de interacción entre agentes

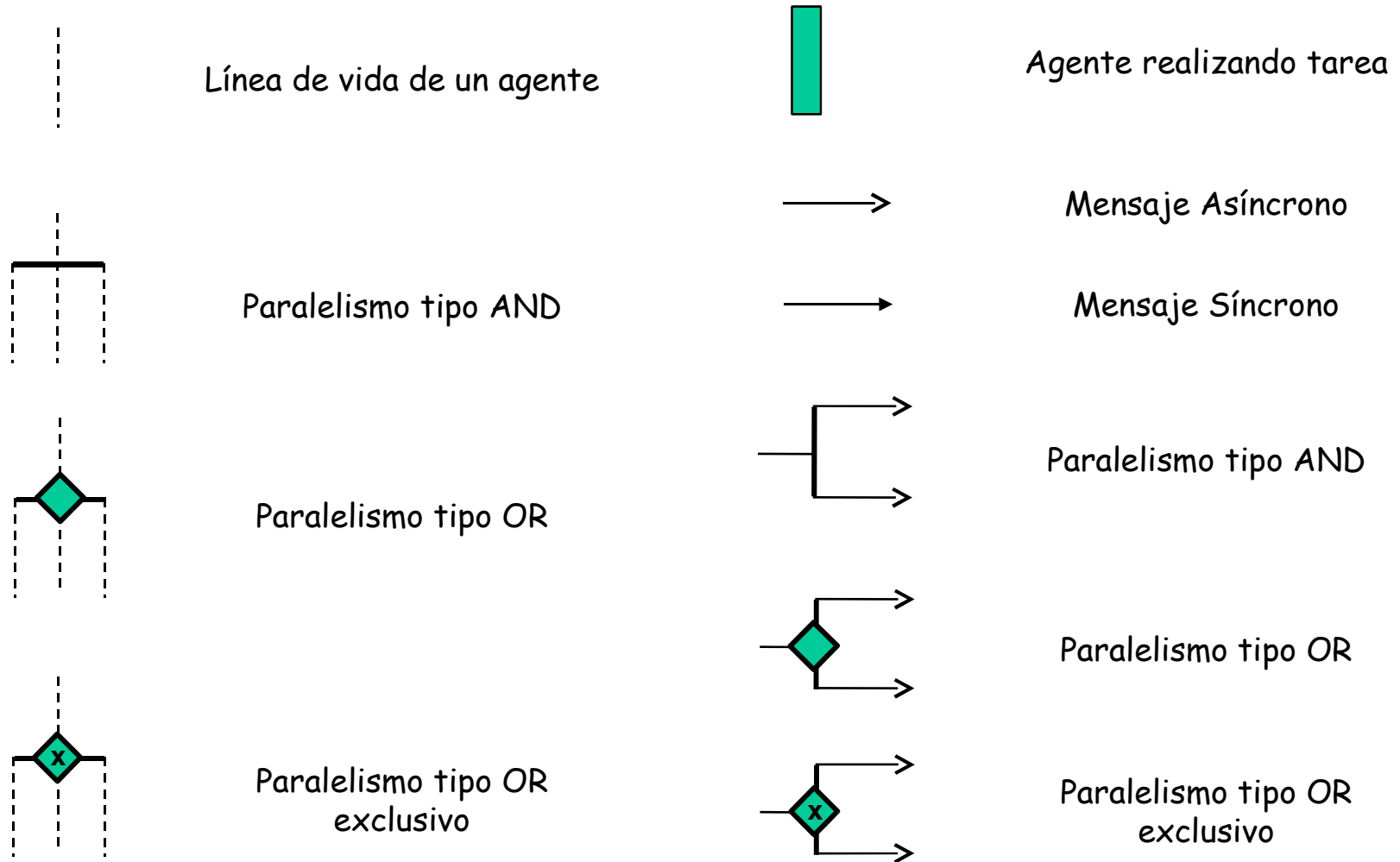
⇒ Diagramas de Protocolo

- ✘ Similar a los diagramas de interacción utilizados en UML

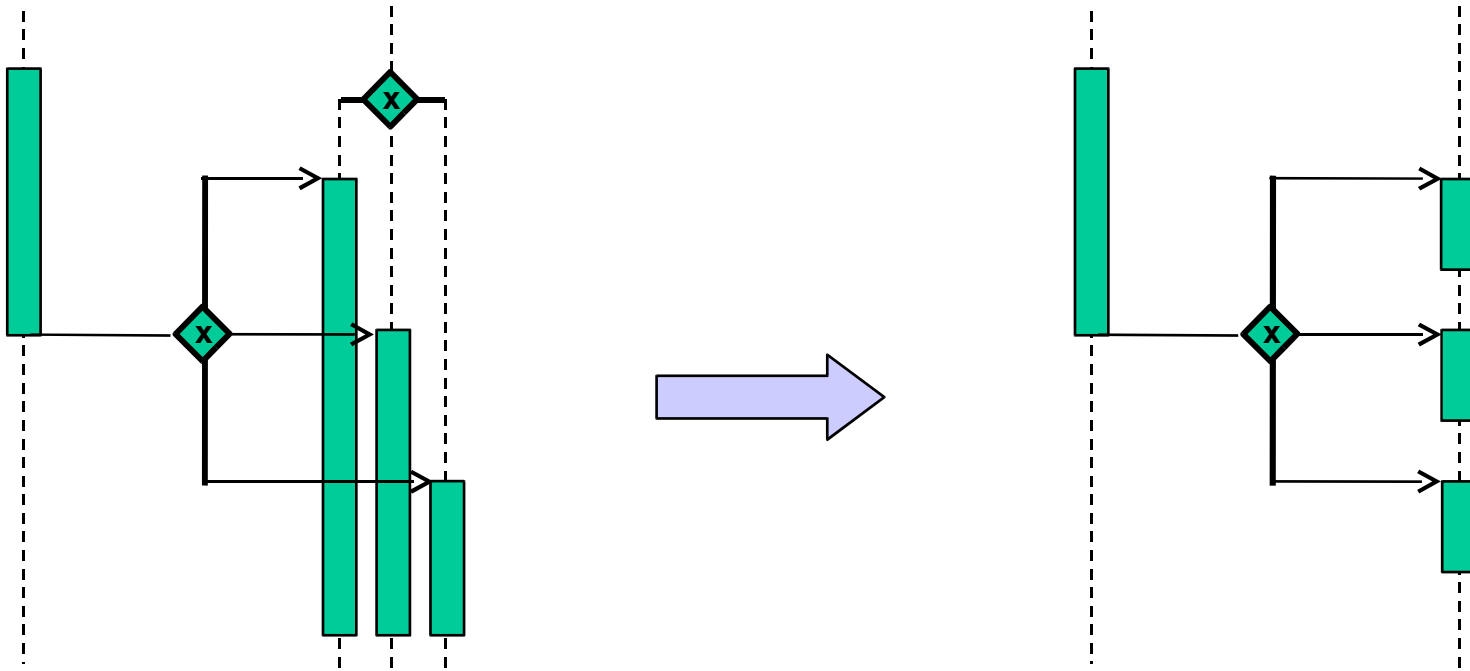
Diagramas de Protocolo

- ⇒ Un Diagrama de Protocolo (DP) tiene 2 dimensiones
 - ✖ La vertical representa el tiempo
 - ✖ La horizontal representa el rol de los agentes involucrados en la interacción
- ⇒ Un DP especifica la secuencia de las comunicaciones por lo que contiene una colección de mensajes parcialmente ordenados entre un agente emisor y un agente receptor
- ⇒ Un DP también permite especificar protocolos de interacción anidados

Elementos básicos de un DP

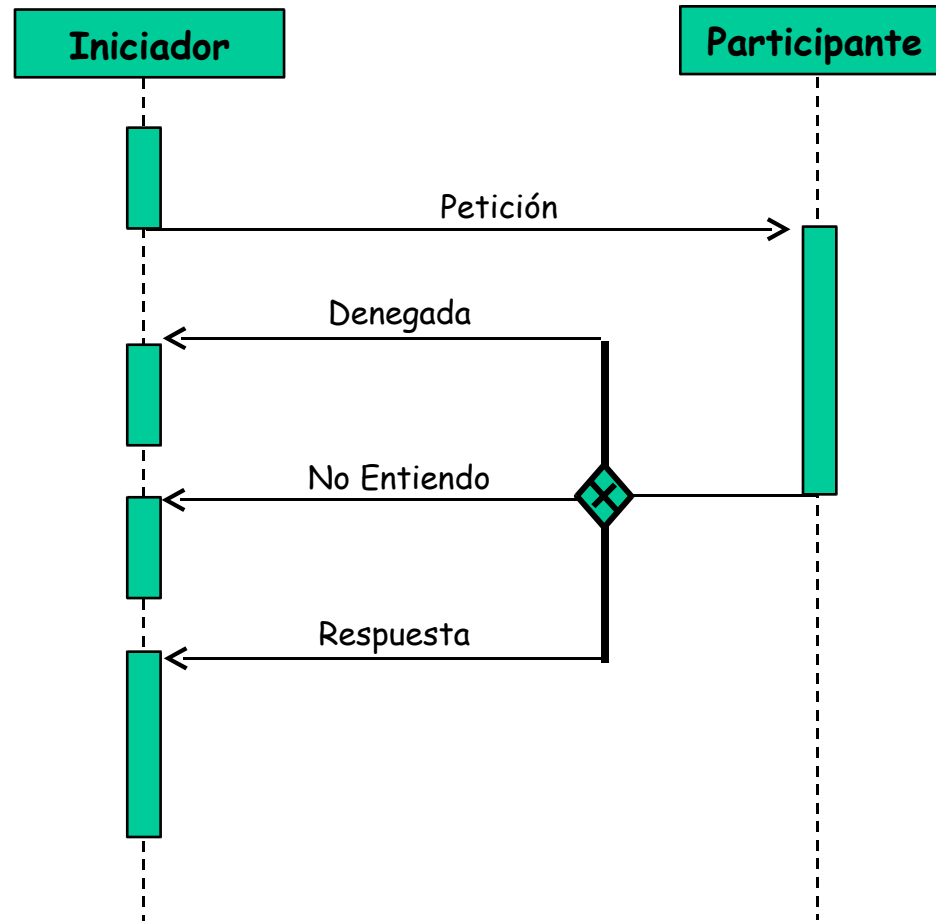


Elementos básicos de un DP

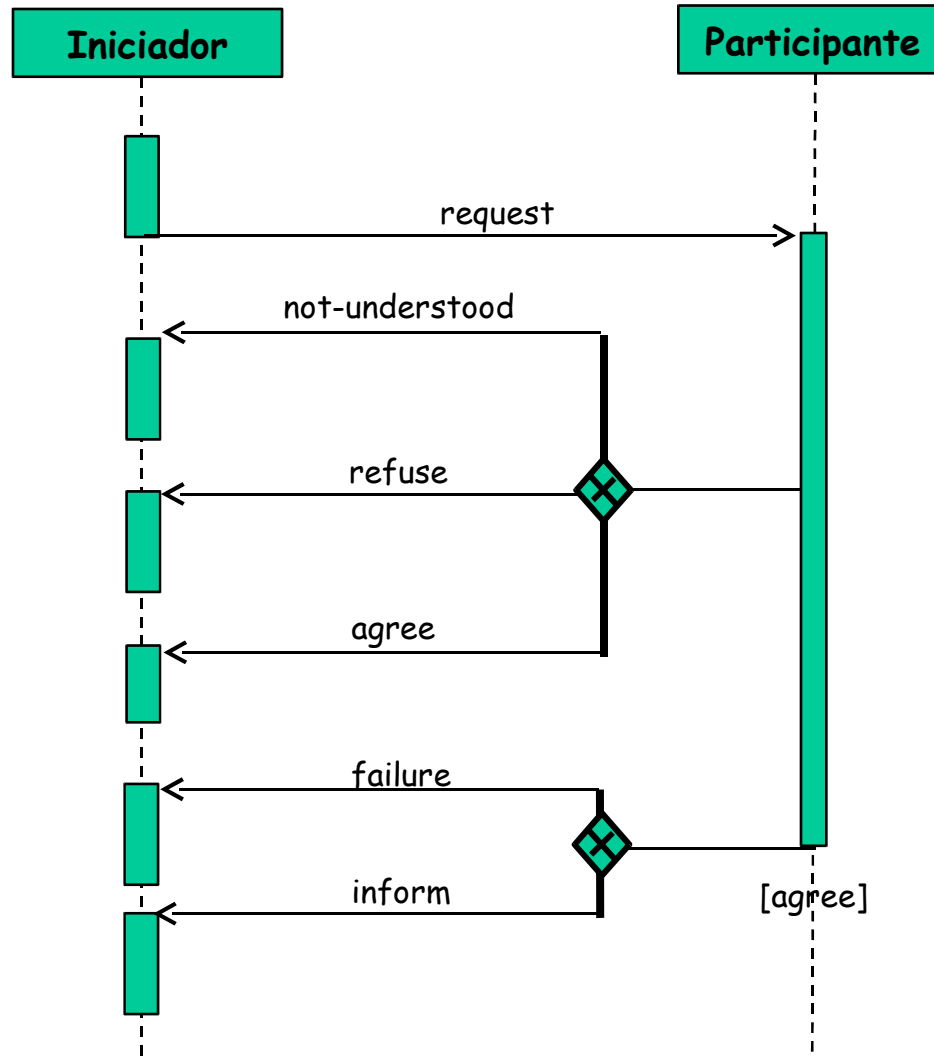


Cuando no hay lugar a confusión se puede abreviar el DP

Diagramas de Protocolo - Ejemplo



FIPA-Request



FIPA-Request (ejemplo)

A envía a B:

(request :content (action (enviar paquete33 Sevilla)) :
protocol FIPA-Request :reply-with q1)

Si no entiende el mensaje, B contesta a A:

(not-understood :
content ((request
:content (action (enviar paquete33 Sevilla))
:protocol FIPA-Request :reply-with q1)
(unknow (word paquete33))) :
protocol FIPA-Request :in-reply-to q1)

FIPA-Request (ejemplo)

Si no puede realizar la acción, B contesta a A:

(refuse :

content ((request

:content (action (enviar paquete33 Sevilla))

:protocol FIPA-Request **:reply-with** q1)

(out-of-service Sevilla) :

protocol FIPA-Request **:in-reply-to** q1)

FIPA-Request (ejemplo)

Si puede realizar la acción, B contesta a A:

```
(agree  
content ((request  
  :content (action (enviar paquete33 Sevilla))  
  :protocol FIPA-Request :reply-with q1  
  (salida paquete33 18:00 (dirección Sevilla)))  
protocol FIPA-Request :in-reply-to q1
```

FIPA-Request (ejemplo)

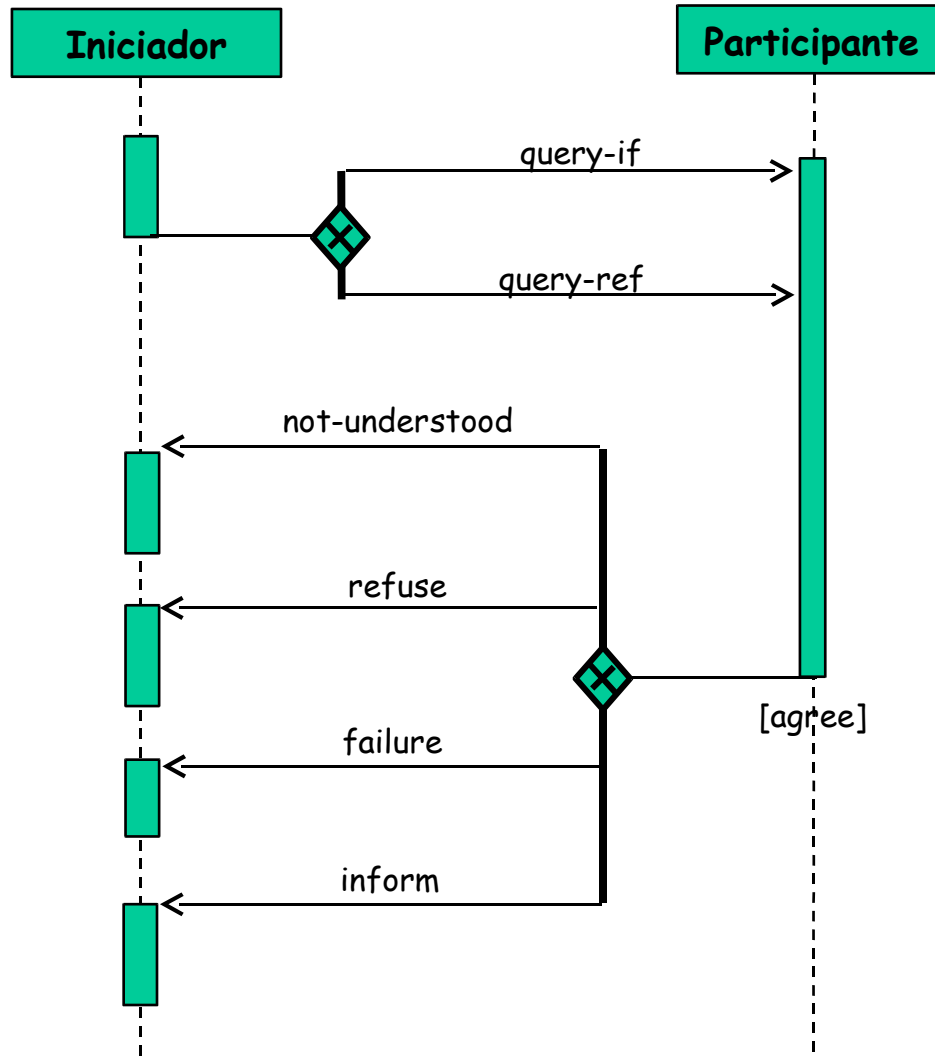
Si ha habido algún error, B contesta a A:

```
(failure  
content ((request  
          :content (action (enviar paquete33 Sevilla)))  
          (error xxxxx))) :protocol FIPA-  
Request :in-reply-to q1)
```

Si la acción se ha ejecutado correctamente, B contesta a A:

```
(inform  
content ((request  
          :content (action (enviar paquete33 Sevilla)))  
          (paquete33 in Sevilla))) :  
protocol FIPA-Request :in-reply-to q1)
```

FIPA-Query



Este protocolo posibilita la petición de información

Se inicia con uno de las siguientes actos comunicativos:

- `query-if` (verdad o falsedad de una proposición)
- `query-ref` (información en referencia a un objeto)

El participante puede responder con:

- `inform` (la información solicitada)
- `refuse` (rechazar la petición)
- `failure` (error)
- `not-understood` (no entiende la petición)

FIPA-Query (ejemplo query-if)

El agente A pregunta a B si está registrado en un servidor ANS:

(query-if :sender A :receiver B
:content (registered (server ans1) (agent B)) :protocol
FIPA-Query :reply-with q1)

El agente B le contesta que no:

(inform :sender B :receiver A
:content (not (registered (server ans1) (agent B)))
:protocol FIPA-Query :in-reply-to q1)

FIPA-Query (ejemplo query-ref)

El agente A pregunta a B por los servicios que tiene disponibles:

(query-ref :sender A :receiver B

:content (all ?x (available-service B ?x)) :protocol

FIPA-Query :reply-with q1)

El agente B le contesta que puede reservar en trenes y aviones:

(inform :sender B :receiver A

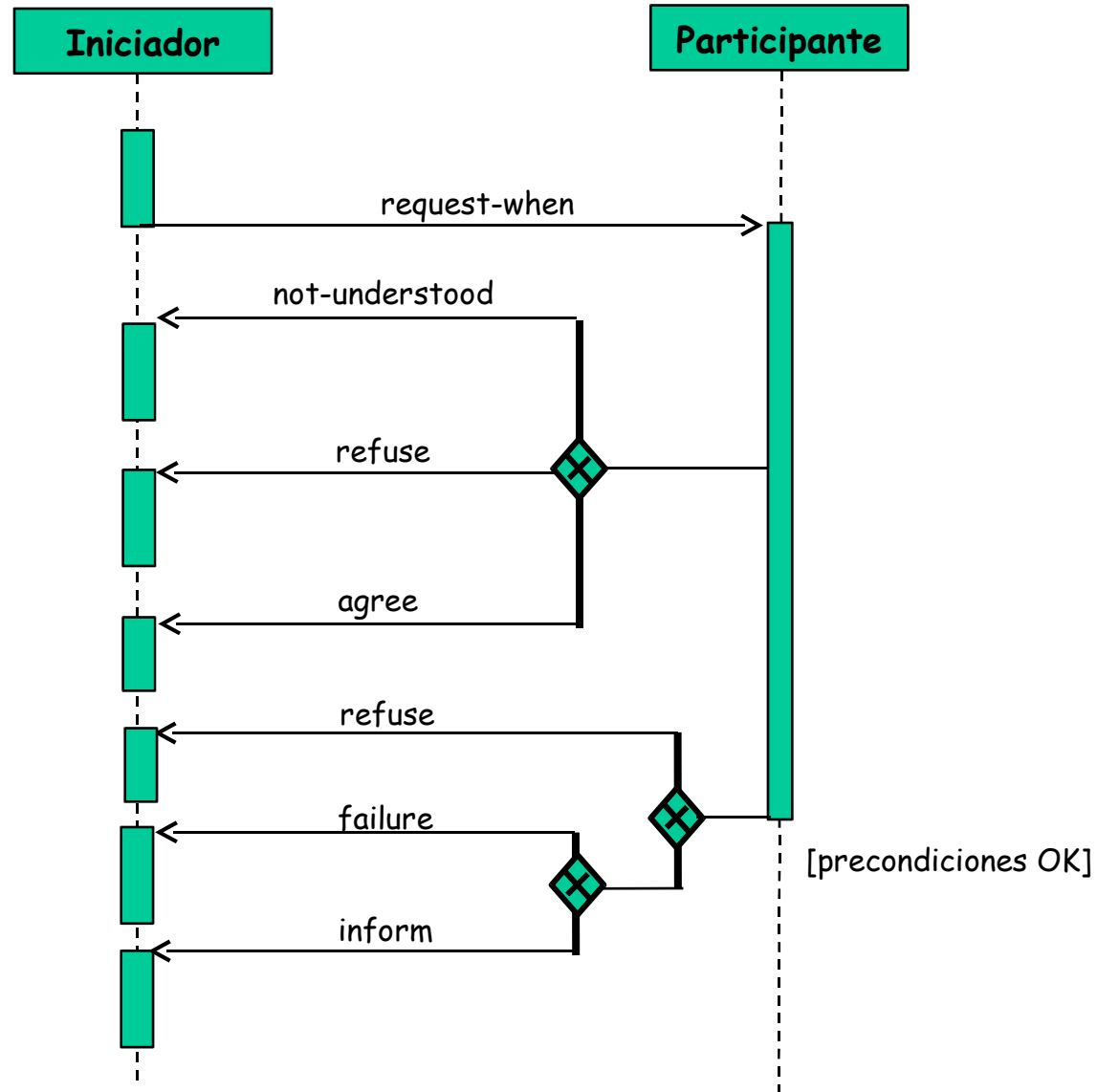
:content (= (all ?x (available-service B ?x))

(set (reserve-ticket train)

(reserve-ticket plane)))

:protocol FIPA-Query :in-reply-to q1)

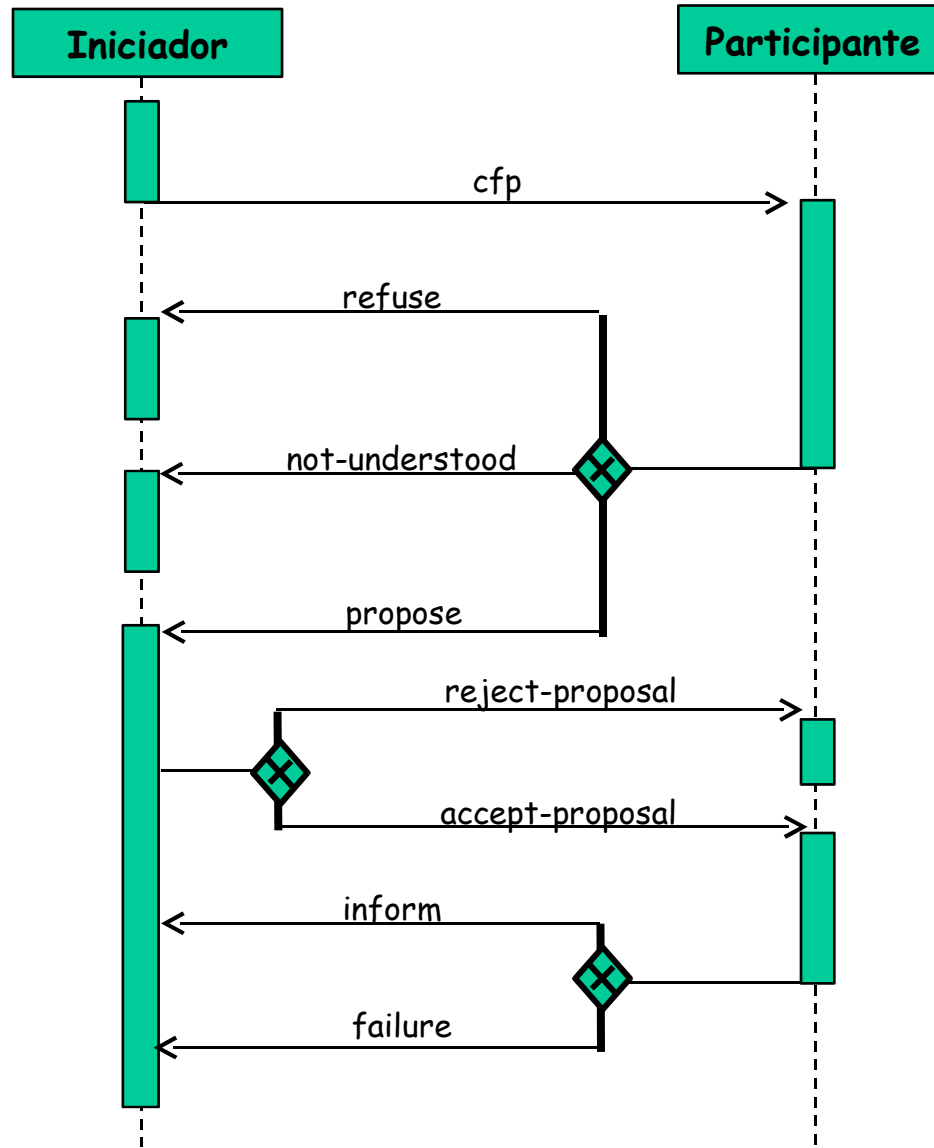
FIPA-Request-When



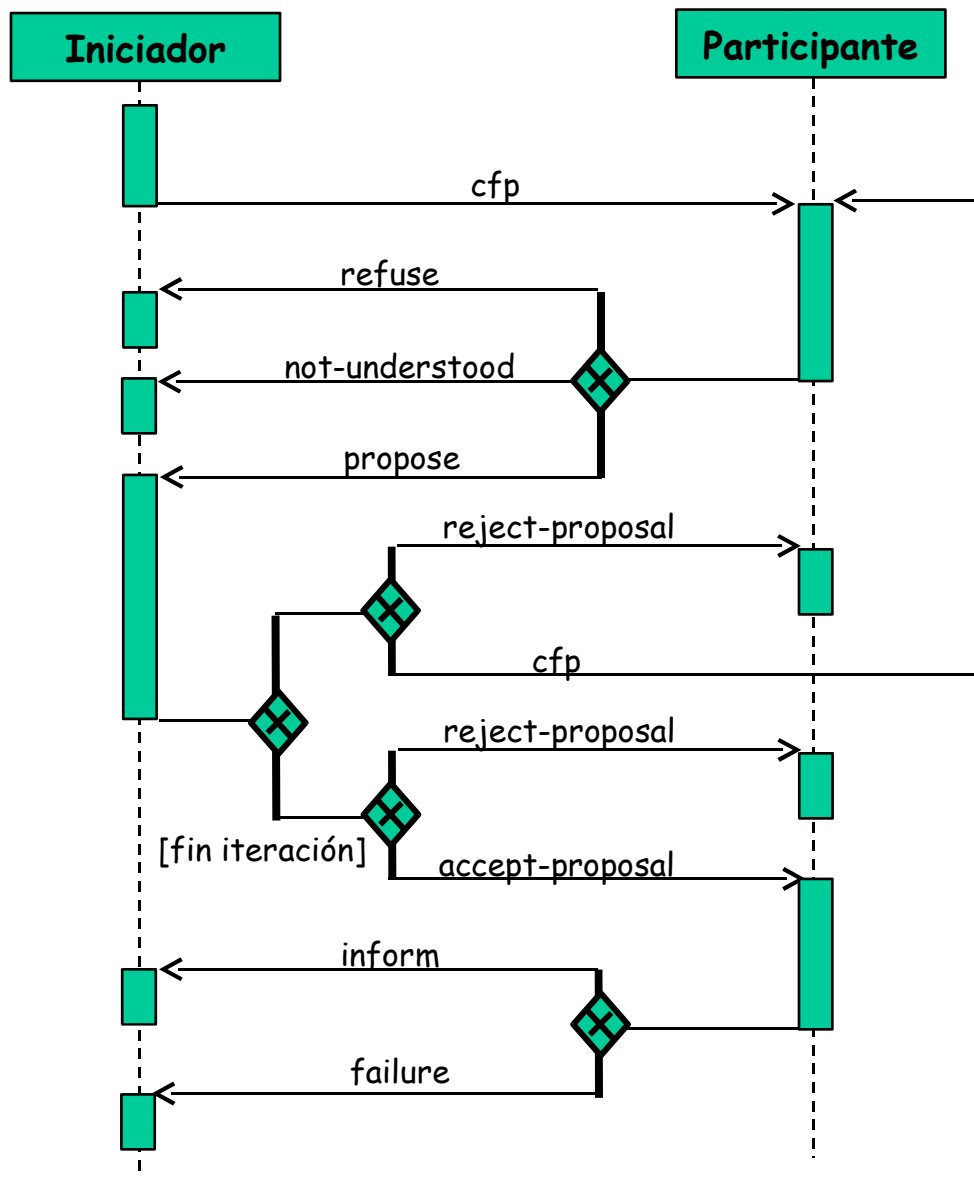
FIPA-Contract-Net

- ⇒ Es un protocolo de Negociación
- ⇒ Este protocolo se utiliza para permitir que un agente iniciador solicite la ejecución de una tarea a uno o más agentes
 - ✗ El iniciador hace una llamada para ver qué agentes pueden realizar la tarea
 - ✗ Los agentes receptores pueden rechazar el mensaje, no entenderlo o devolver al iniciador una propuesta con sus precondiciones para realizar la tarea
 - ✗ Cuando el iniciador recibe las propuestas, las evalúa, escoge a un agente (o a varios) para realizar la tarea y envía una aceptación al agente (o agentes) elegido y un rechazo al resto
 - ✗ Cuando los agentes contratados realicen la tarea informan al iniciador. Estos agentes pueden subcontratar la tarea a otros

FIPA-Contract-Net



FIPA-Iterated-Contract-Net



FIPA-English-Auction

- ⇒ Protocolo de Subasta
- ⇒ El subastador intenta encontrar el precio de venta de un objeto proponiendo inicialmente un precio por debajo del valor del mercado para después ir incrementándolo gradualmente
- ⇒ Una vez anunciado el precio, el subastador espera un tiempo para ver si algún comprador expresa su intención de pagar dicha cantidad
- ⇒ Si un agente acepta el precio, el subastador incrementa el precio y lo anuncia al resto esperando nuevas propuestas
- ⇒ La subasta termina cuando no hay pujas para el último precio propuesto
 - ✗ Se vende si en la última puja el precio supera un determinado precio de reserva (privado para el subastador)
 - ✗ No se vende si en la última puja el precio no supera el precio de reserva

FIPA-Dutch-Auction

- ⇒ Protocolo de Subasta
- ⇒ En este protocolo un subastador intenta encontrar el precio de venta de un objeto proponiendo inicialmente un precio por encima del valor del mercado para después irlo reduciendo gradualmente
- ⇒ Al anunciar el precio, el subastador espera un tiempo para ver si algún comprador expresa su intención de pagar la cantidad
- ⇒ Si ningún agente acepta el precio, el subastador disminuye el precio y lo anuncia al resto esperando nuevas propuestas
- ⇒ La subasta termina cuando
 - ✗ Hay algún comprador para el precio propuesto (se vende)
 - ✗ Se ha bajado el precio hasta el mínimo permitido (no se vende)
- ⇒ Inspirado en las subastas de los mercados de flores holandeses

Ontologías

- ⇒ Modelo de Comunicación basado en la asunción de que 2 agentes que quieren conversar han de compartir una ontología común que describa el universo de discurso
- ⇒ Esto asegura que estos agentes atribuyen el mismo significado a los símbolos utilizados en los mensajes
- ⇒ Una ontología es una descripción formal de los conceptos y relaciones que pueden existir en una determinada comunidad de agentes
- ⇒ Una ontología es una especificación de una conceptualización
 - ✗ Dos ontologías pueden diferir en el vocabulario utilizado (por ejemplo, castellano e inglés) y sin embargo compartir la misma conceptualización

Ontología - Ejemplo (SHOE)

⇒ Identificación

- × <ONTOLOGY ID="SIA-Dpto" VERSION="1.0">

⇒ Categorías

- × <DEF-CATEGORY NAME="Persona" ISA="base.SHOEntity">
- × <DEF-CATEGORY NAME="Profesor" ISA="Persona">
- × <DEF-CATEGORY NAME="Estudiante" ISA="Persona">

⇒ Relaciones

- × <DEF-RELATION NAME="Tutor de">
- × <DEF-ARG POS="1" TYPE="Profesor">
- × <DEF-ARG POS="2" TYPE="Estudiante">
- × </DEF-RELATION>

- × <DEF-RELATION NAME="Edad">
- × <DEF-ARG POS="1" TYPE="Persona">
- × <DEF-ARG POS="2" TYPE=".NUMBER">
- × </DEF-RELATION>

Metodología de Construcción de Ontologías

⇒ Especificación

- ✖ Propósito de la ontología
- ✖ Ámbito de la ontología
- ✖ Fuentes de conocimiento

⇒ Conceptualización

- ✖ Utilizar las fuentes de conocimiento para refinar la lista de términos y sus significados
 - ☒ Expertos, Libros, Brainstorming, Entrevistas,...

⇒ Formalización

- ✖ Descripción de los conceptos y sus relaciones en un modelo estándar

Especificación - Ejemplo

Dominio: Química

Fecha: 23 – 10 – 2002

Autor: Productos Químicos, S. A.

Propósito:

Ontología relativa a las sustancias químicas utilizadas cuando se requiere información para la enseñanza, fabricación y análisis de elementos químicos. Esta ontología se utilizará para consultar, por ejemplo, el peso atómico del Sodio

Ámbito:

Lista de 103 elementos químicos: Litio, Sodio, Cloro,...

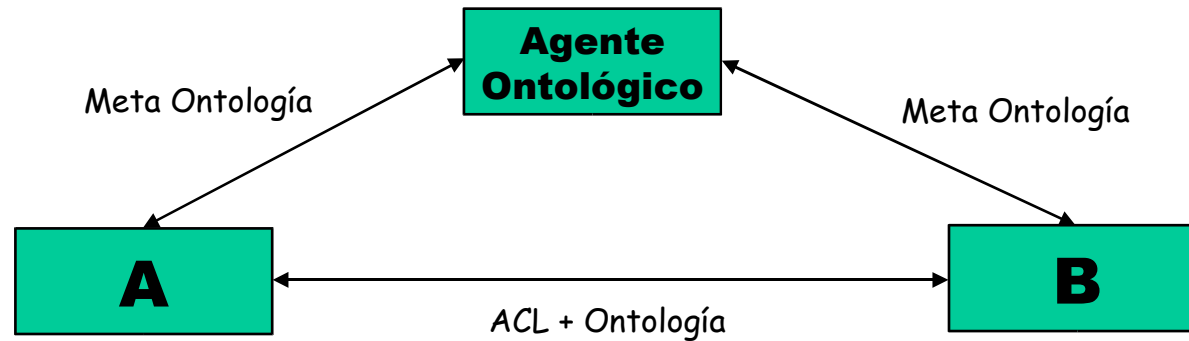
Lista de conceptos: Halógenos, Gases nobles, Metales,...

Lista de propiedades y valores: Número atómico, Peso atómico, Volumen atómico a 20° C,...

Fuentes de Conocimiento:

Handbook of Chemistry and Physics. 65th edition. CRC-Press Inc., 1985.

Agentes Ontológicos



- ⇒ Los agentes pueden resolver problemas de entendimiento en tiempo de ejecución
 - ✗ Empleando una Meta Ontología para hablar sobre ontologías con un agente ontológico
- ⇒ El agente ontológico realiza las siguientes tareas
 - ✗ Crear y mantener ontologías (o sólo algunos términos de las ontologías)
 - ✗ Traducir expresiones entre distintas ontologías (diferentes nombres con el mismo significado)
 - ✗ Comprobar si dos ontologías son idénticas (hacen referencia a una misma conceptualización) o si son equivalentes en algún subconjunto de conceptos
 - ✗ Ayudar a los agentes a localizar ontologías