

Introducción a los Sistemas Multi-Agente

Sistemas Multi-Agente y Sistemas Autónomos

Juan A. Botía

Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia

October 3, 2007

Sistemas multi-agente

- En algunos casos los agentes se pueden desenvolver de forma aislada
- El caso de un único agente representando a un sistema completo y sin interacción con otros agentes de su entorno es cada vez más raro.
- Razón principal: interacción creciente entre las máquinas que están conectadas a Internet.
- Razón histórica: evolución de la IAD (Inteligencia Artificial Distribuida)

Justificación histórica

Los sistemas tradicionales de IAD están compuestos por

- módulos muy especializados y fuertemente dependientes de conocimiento
- interacciones entre módulos rígidas y perfectamente definidas

Problemas:

- En el desarrollo de sistemas expertos complejos, el conocimiento estaba distribuido entre un equipo de personas
 - ▶ La suma de conocimientos individuales \neq conocimiento del grupo
 - ▶ Necesario modelar procesos de
 - ★ Intercambio de experiencias,
 - ★ ajuste en los métodos,
 - ★ negociación
- Naturaleza distribuida de los problemas
 - ▶ Robots exploradores (+) (cooperación)
 - ▶ Controladores aéreos (-) (competición)

Planteamiento general de los SMAs

- Habilitar interacción entre agentes, siguiendo reglas para
 - ▶ cooperación,
 - ▶ resolución de conflictos y
 - ▶ competición
- mientras que todos los agentes del sistema tratan de sobrevivir para resolver problemas

SMAs: definición (Ferber, 1999 [1])

Definición

El término sistema multi-agente (o SMA) se refiere a sistemas que están compuestos por los siguientes elementos:

- 1 *Un entorno, E , que es un espacio que, generalmente, tiene un volumen.*
- 2 *Un conjunto de objetos O . Estos objetos están situados (i.e. tienen asociada una posición en E). Además alguno de estos objetos son pasivos lo que implica que pueden ser percibidos, creados, destruidos y modificados por los agentes.*
- 3 *Un conjunto de agentes, A , que son también objetos ($A \subseteq O$), y representan las entidades activas del sistema.*
- 4 *Un conjunto de relaciones R , que enlaza objetos en O .*
- 5 *Un conjunto de operaciones, Op , que hacen posible para los agentes de A el percibir, producir, consumir, transformar y manipular objetos pasivos de O .*
- 6 *Una relación de operadores que representan la aplicación de las operaciones en Op y al reacción del mundo a este intento de modificación, que se denominarán las leyes del universo.*

SMA: tipos

- Cuando $A = O$ y $E = \emptyset$ entonces R es un grafo, cada nodo un agente y cada arco una relación y el SMA se denomina comunicante puro
 - ▶ Actos comunicantes con intención
 - ▶ Comunicación mediante paso de mensajes entre los agentes.

Definición

Un agente software comunicante puro es una entidad computacional que

- 1 *es un sistema de computación abierto (conjunto de aplicaciones, redes y sistemas heterogéneos),*
- 2 *puede comunicarse con otros agentes,*
- 3 *está dirigido por un conjunto de objetivos propios,*
- 4 *tiene recursos propios,*
- 5 *tiene una representación parcial del resto de agentes,*
- 6 *tiene servicios que puede ofrecer al resto de agentes,*
- 7 *manifiesta un comportamiento global dirigido a la consecución de sus objetivos, teniendo en cuenta los recursos y habilidades (servicios) disponibles y dependiendo en sus modelos internos y la comunicación que recibe.*

SMA: tipos (II)

Cuando

- E es un espacio métrico
- los agentes son capaces de reconocer los objetos situados en E y
- capaces de actuar en el entorno (y por ende modificar el estado del sistema)

entonces el SMA se dice que es *situado*. Un SMA es situado puro si la comunicación no se usan mensajes sino la propagación de señales generadas a partir de la modificación del entorno.

Agente software situado puro

Definición

Un agente software situado puro es una entidad física (o quizás computacional si es simulada) que

- 1 *está situada en su entorno,*
- 2 *está gobernada por una función de supervivencia/satisfacción,*
- 3 *tiene recursos propios en forma de herramientas o baterías,*
- 4 *es capaz de percibir su entorno (con una capacidad limitada),*
- 5 *no posee casi representación de su entorno,*
- 6 *tiene habilidades,*
- 7 *manifiesta un comportamiento global dirigido a la supervivencia o la optimización de una función de satisfacción, teniendo en cuenta los recursos, percepciones y habilidades disponibles.*

Niveles de organización en los SMAs

- 1 El nivel micro-social: incluye
 - ▶ interacciones entre agentes individuales,
 - ▶ tipos de enlaces existentes entre dos o más agentes
- 2 El nivel grupal:
 - ▶ las estructuras organizativas intermediarias que participan en la composición de organizaciones
 - ▶ la diferenciación de roles entre los distintos agentes,
 - ▶ emergencia de estructuras organizativas entre agentes y
 - ▶ la agregación de nuevos componentes durante la constitución de las organizaciones.
- 3 El nivel de sociedades globales o poblaciones:
 - ▶ evolución y dinámica de grupos de un gran número de agentes
 - ▶ estructura organizativa a nivel global.

Comunicación entre agentes

- ¿Qué servicios se asumen?
 - ▶ Se asume siempre un servicio de transporte de mensajes de agentes robusto y sincronizado
- ¿Podemos plantearla al nivel OO?
 - ▶ Sean dos objetos Java o_1 y o_2 .
 - ▶ El objeto o_1 tiene un método m que el objeto o_2 ha de invocar.
 - ▶ La comunicación entre o_1 y o_2 se produce a través de la invocación del método
 - ▶ ¿quién decide sobre la necesidad de la invocación? o_2
 - ▶ ¡ o_1 no tiene autoridad sobre su propio método!

La autonomía condiciona la comunicación

Vamos a plantear el mismo ejemplo con agentes

- Sean dos agentes i y j ,
- i tiene la capacidad de realizar la acción α (equivalente a la ejecución de un método)
- Pero la idea “agente j invoca la acción α en el agente i ” no tiene sentido
- Si el agente j invocara de alguna forma la acción α en i , no tendría ninguna garantía de su efectividad
- El agente i es el que decide, en base a sus intereses, si ha de realizar la acción y en qué condiciones.

Comunicación mediante influencia

- En general, un agente no puede obligar a realizar acciones, ni directamente modificar el estado interno de otros agentes.
- La comunicación se hace influyendo en los otros
- Si yo digo

“Está lloviendo en Murcia”

intento influir al receptor a que crea lo mismo que yo.

- El receptor podrá creerlo, o no, dependiendo de la credibilidad que yo tenga con respecto a él
- La teoría de los actos del habla trata la comunicación como si fueran acciones
- Acciones, generadas a partir de intenciones

Actos del habla

- Se debe a John Austin.
 - ▶ Percibió que algunas sentencias del lenguaje natural podían considerarse como acciones en tanto que cambian el estado del mundo, e.g. “Yo os declaro Marido y Mujer”
- Identificó un conjunto de verbos denominados *performative verbs* y que nosotros denominaremos performativas, como exigir (*request*), informar y prometer.
- Los actos del habla están formados por tres componentes:
 - ▶ El componente locutorio (materia)
 - ▶ El componente ilocutorio (acción)
 - ★ Fuerza ilocutoria (e.g. afirmación, pregunta, petición de acción, promesa, orden, información, etc)
 - ★ Contenido proposicional
 - ▶ El componente perlocutorio (efectos de la ilocución)

Actos del habla e IA

- Cohen y Perrault lo usaron primero para la IA
 - ▶ Si un agente va a comunicarse con otros, sus planes deben incluir actos del habla (representación de performativas usando la notación de STRIPS con pre y postcondiciones)
- Ejemplo: performativa *request* (intención es conseguir que el escuchante realice una acción).
- Dos tipos de precondiciones:
 - ▶ *can-do*
 - ★ hablante debe tener la creencia de que el escuchante es capaz de realizar la acción
 - ★ hablante debe creer que el escuchante también cree que él mismo es capaz de realizarla
 - ▶ *want*
 - ★ el hablante debe realmente creer que él quiere que la acción *Request* se lleve a cabo

Actos del habla e IA

- Sea $Request(H, E, \alpha)$ el acto del habla para un hablante H , un escuchante E y una acción a realizar α .
- Precondiciones 1 y 2

$$(H \text{ BELIEVE } (E \text{ CANDO } \alpha)) \wedge \\ (H \text{ BELIEVE } (E \text{ BELIEVE } (E \text{ CANDO } \alpha))).$$

- Precondición 3

$$(E \text{ BELIEVE } (E \text{ WANT } requestInstance)).$$

- Postcondición: el hablante tiene la creencia de que el escuchante cree que que el hablante quiere que la acción α se realice.

$$E \text{ BELIEVE } (H \text{ BELIEVE } (H \text{ BELIEVE } \text{WANT } \alpha))$$

ACLs: KQML

- El primer mecanismo de comunicación extendido
- A partir de DARPA se creó la iniciativa *Knowledge Sharing Effort* dentro de la cual se crearon KQML (*Knowledge Query and Manipulation Language*) [2] y KIF (*Knowledge Interchange Format*).
- KQML es un ACL (*Agent Communication Language*)
- KIF es un lenguaje para representación del contenido del mensaje

KQML: presunciones iniciales

Son las siguientes [2] :

- Los agentes están conectados mediante enlaces de comunicación unidireccionales que transportan mensajes. Además, estos enlaces conllevan un retraso en el transporte mayor que cero.
- Cuando el agente recibe un mensaje, sabe desde que enlace le llega. Por otro lado, cuando envía un mensaje, conoce el link a través del que sale.
- Los mensajes enviados al mismo destino llegan en el mismo orden en el que fueron enviados.
- El envío de mensajes es fiable.

KQML: sintaxis

- Sintaxis inspirada en LISP y tiene la forma

```
(<word> {<whitespace> : <word> <whitespace>
<expression>}*)
```

- El primer <word> es la performativa
- La construcción interior es la especificación de parámetros y sus valores.

<i>Parámetro</i>	<i>Significado</i>
:content	El contenido del mensaje KQML
:force	El agente puede indicar si mantendrá su creencia en la performativa de forma permanente
:language	El lenguaje con el que se expresa el valor de :content
:ontology	El cuerpo ontológico asociado a :content
:receiver	El receptor del mensaje
:reply-with	Al recibir la réplica (si se espera) al mensaje, el receptor deberá responder con la misma
:in-reply-to	Si el mensaje es una réplica, la etiqueta con la que se esperaba la misma

KQML, performativas

- En [2] se incluye un total de 41 performativas
 - ▶ para informar a otros agentes de hechos que el agente emisor conoce (`tell`, `untell`),
 - ▶ para influir insertando, borrando hechos en la base de conocimiento del receptor (`insert`, `delete`),
 - ▶ respuestas básicas de error (`error`, `sorry`),
 - ▶ para realizar preguntas (`evaluate`, `reply`, `ask-if`, ...), etc.

KQML, ejemplo

Un ejemplo de conversación es el ejemplo siguiente, que puede consultarse en [2], pág.5. Un agente A desea que el agente B evalúe determinada expresión:

```
(evaluate :language KIF :ontology motors  
:reply-with q1  
:content (var (torque motor1) (sim-time 5)))
```

Y el agente B responde con:

```
(reply :language KIF :ontology motors :in-reply-to  
q1 :content (scalar 12 kgf))
```

Se producen críticas a la ambigüedad, a performativas que no lo són y por falta de performativas

- Críticas a la ambigüedad

- ▶ deny de una tell: ¿significa entonces que el agente no envió un tell antes o que no cree lo que se está diciendo ahora?
- ▶ Lo que el deny está negando es un mensaje anterior → los mensajes cobran un valor de verdad y se convierten en proposiciones
- ▶ Solución: propuesta del 97 [7], que evita incluir mensajes en el cuerpo de estas performativas

- Performativas que, en realidad, no lo son
 - ▶ La performativa `achieve` significa que el emisor quiere que una proposición se haga verdad en el receptor
 - ▶ Crítica: el emisor no puede ejecutar acciones en el receptor
 - ▶ Solución: usar la performativa `request` con `achieve` en el cuerpo del mensaje
 - ▶ No se tomó en cuenta en la propuesta del 97

- Performativas que faltan (ver [9])
 - ▶ Faltan performativas para expresar compromiso con realizar una acción (i.e. `promise`, `agree`)
 - ▶ Si un agente no puede expresar compromiso es imposible construir SMAs robustos
 - ▶ Tampoco se tuvieron en cuenta en la especificación del 97

Conversaciones entre agentes

- Los actos comunicativos simples no son conversaciones
- Un acto comunicativo es bien el originador de una conversación, bien la consecuencia de un acto comunicativo previo
- Por ejemplo, cuando un agente emite un request, esperará que se le responda si se acepta o no realizar la acción y, posiblemente, una notificación de que la acción ha sido realizada.
- Una conversación proporciona el potencial para la coordinación, cooperación, negociación
- Necesidad de modelar las conversaciones porque
 - ▶ Si se es capaz de modelar internamente los diálogos, el comportamiento será el correcto
 - ▶ Por otro lado, sus actitudes mentales evolucionarán adecuadamente

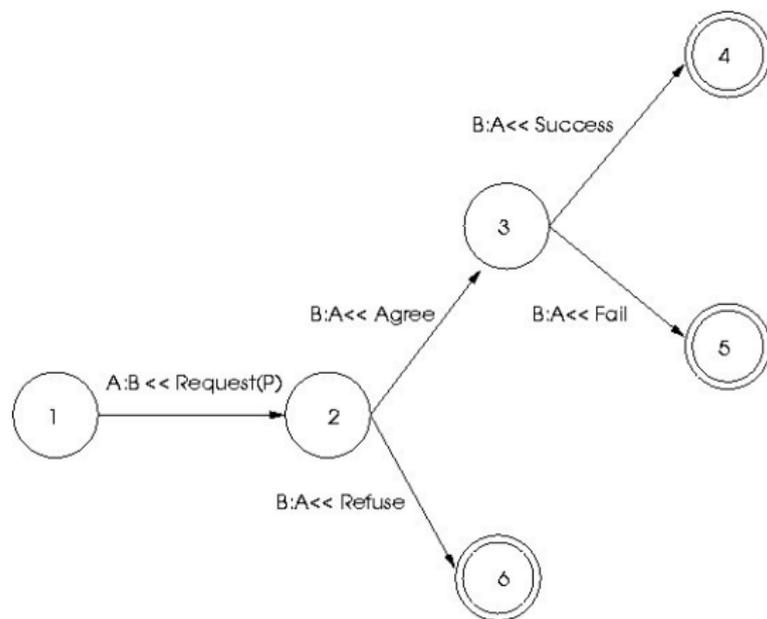
Modelado de conversaciones mediante AFDs

- Notación: cuando el agente A envía un mensaje al agente B, y que el mensaje corresponde a la performativa `performative` y al contenido `content` lo denotamos con

`A : B << Performative(content)`

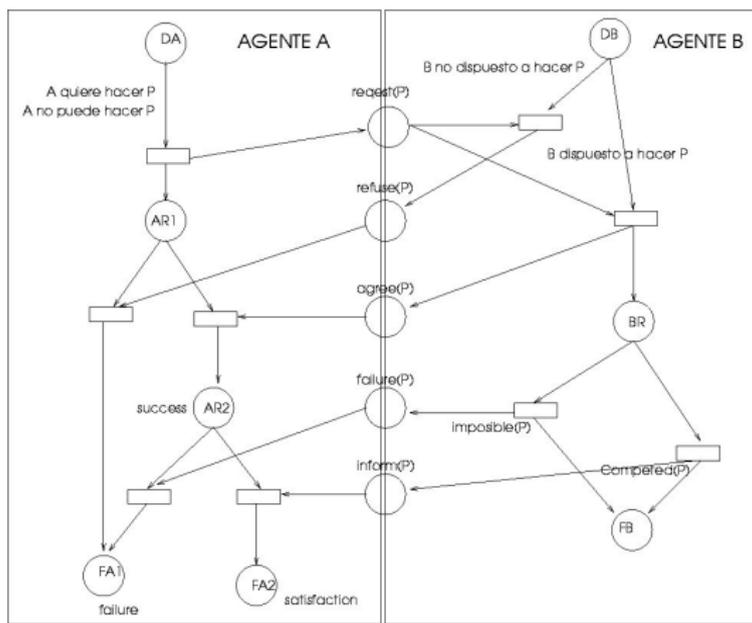
- Modelado mediante AFDs
 - ▶ Los estados del autómatas son estados posibles en la comunicación.
 - ▶ Las transiciones equivalen a realización de actos comunicativos
 - ▶ Los estados finales del autómatas equivalen a posibles formas de finalizar la conversación.

Modelado de conversaciones mediante AFDs



Modelado mediante redes de Petri

- Con redes de Petri vamos a poder incluir en la conversación los estados mentales del agente como marcas de activación



¿Para qué usar un protocolo de interacción?

- Porque los agentes de un SMA actúan guiados por la consecución de sus objetivos individuales (*self-interested*).
- también porque existen situaciones en las que comparten objetivos comunes (*benevolent*)

Los protocolos de interacción del primer caso están orientados a maximizar los valores de la función de utilidad generados por las acciones de los agentes.

En el segundo caso se enfrentan a

- 1 trabajar con objetivos comunes,
- 2 trabajar con tareas comunes,
- 3 evitar conflictos en la medida de lo posible y
- 4 mantener un flujo correcto de conocimiento y evidencias

Coordinación en SMAs

Definición

La coordinación en SMAs trata de cómo los agentes se comportan individual y socialmente para que, por un lado, se satisfagan los objetivos personales y, por el otro, los globales.

¿Para qué?

- Los recursos en un SMA son limitados
- La expertise está repartida entre los agentes
- Evitar el caos (anarquía) [3]
- Cumplir un cjto. de restricciones globales
- Cumplir roles especializados
- Interdependencia entre objetivos de unos y acciones de otros
- Por eficiencia

Mecanismos para la coordinación

La coordinación se consigue mediante la distribución de

- los datos y (visión parcial de la totalidad)
- el control (autonomía en agentes)

con lo que la coherencia global es **difícil**

Para atacar el problema de la coordinación necesitamos modelar, en la fase de diseño, las posibles interdependencias que puedan haber entre agentes, sus acciones y objetivos.

Grafos AND/OR de tareas y objetivos

- Representa las dependencias entre objetivos y tareas que se necesitan para cumplir objetivos primitivos (que son los nodos hoja del árbol [4, 5])
- Algunas suposiciones
 - ▶ Sean dos agentes, $Agente_1$ y $Agente_2$
 - ▶ Han de cumplir los objetivos G_0^1 y G_0^2
 - ▶ Sea el grafo de interdependencias el que aparece seguidamente

Grafo aumentado de $Agente_1$ y $Agente_2$

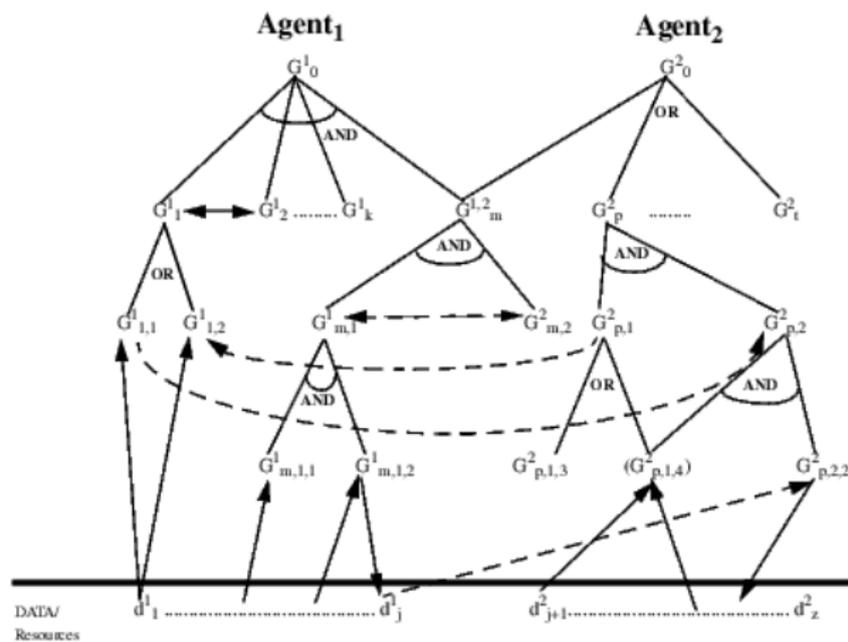


Figure 1: A distributed goal search tree involving $Agente_1$ and $Agente_2$. The dotted arrows indicate interdependencies between goals and data in different agents, solid arrows dependencies within an agent. The superscripts associated with goals and data indicate the agent which contains them (Jennings, 1993).

Interdependencias entre $Agente_1$ y $Agente_2$

- $G_{n_1, n_2, \dots, n_m}^i$ es un subobjetivo del agente i , hijo n_m -ésimo de $G_{n_1, n_2, \dots, n_{m-1}}^i$
- El cjto. de superíndices a_1, a_2, \dots, a_n en un objetivo indica que los agentes con esos índices deben cumplir todos el objetivo
- Interdependencias
 - ▶ Entre G_1^1 y G_2^1
 - ▶ Entre $G_{1,1}^1$ y $G_{p,2}^2$
 - ▶ G_1^1 y G_p^2 son interdependientes si $G_{1,1}^1$ se resuelve para resolver así G_1^1 .
 - ▶ Interdependencias indirectas por producción y consumo del mismo recurso: $G_{m,1,2}^1$, $G_{p,2,2}^2$ y d_j^1 .

Clasificación de interdependencias

- Débiles o fuertes
 - ▶ Dependiendo de si se debe satisfacer necesariamente para que el objetivo se cumpla.
- Unidireccional o bidireccional
 - ▶ Dependiendo del sentido de la dependencia
 - ▶ $G_{1,1}^1 \rightarrow G_{p,2}^2$ significa que el objetivo $G_{p,2}^2$ depende de $G_{1,1}^1$
 - ▶ $G_{m,1}^1 \leftrightarrow G_{m,2}^2$ significa que los objetivos de los dos agentes se ven afectados

Coordinación a partir del grafo

La naturaleza de las interdependencias determina el tipo de coordinación

- 1 El agente $Agent_1$ sabe que el objetivo $G_{p,2,2}^2$ necesita el recurso d_j^1 para empezar (es una dependencia fuerte y unidireccional)
 - ▶ Podría decidir ejecutar $G_{m,1,2}$ antes que $G_{m,1,1}$.
- 2 Dados $G_{m,1}^1 \leftrightarrow G_{m,2}^2$,
 - ▶ los dos agentes podrían establecer un proceso de negociación para acordar sus respectivos tiempos de ejecución.
- 3 Ai el agente $Agent_1$ escoge $G_{1,1}^1$ para satisfacer G_1^1 , el cumplimiento del objetivo podría suministrar información relevante para que el agente $Agent_2$ pudiera realizar el objetivo $G_{p,2}^2$
 - ▶ El agente $Agent_1$ podría compartir la información generada por $G_{1,1}^1$ con el agente $Agent_2$ para resolver eficientemente $G_{p,2}^2$

Construcción del grafo

Para construir un árbol de este tipo, los pasos son los siguientes:

- 1 definir el grafo de objetivos, incluyendo la identificación y la clasificación de dependencias (i.e. si son del tipo AND o del tipo OR).
- 2 Asignar regiones particulares del grafo a los agentes apropiados.
- 3 Controlar las decisiones acerca de qué áreas del grafo explorar.
- 4 Recorrer el grafo.
- 5 Informar de los recorridos con éxito.

Algunas de las actividades serán cooperativas y otras serán realizadas por un solo agente. Esto último se decide en la fase de diseño.

Cooperación en SMAs

Definición

La cooperación en un SMA consiste en la actuación coordinada entre agentes de tal manera que unos colaboran en la resolución de tareas de otros interesada o desinteresadamente.

Puede existir cooperación entre agentes benevolentes y egoistas

- Los primeros cooperan de forma natural al resolver un problema global, cada uno dedicado a su *parcela*
- Los segundos necesitan un proceso negociador para determinar si es interesante o no cooperar y en qué términos

Descomposición de tareas

Previo a toda negociación se necesita descomponer

- ¿Cómo hacerlo?
 - ▶ en el diseño
 - ▶ mediante planificación jerárquica
 - ▶ inherente al problema
- Luego se pueden distribuir según los criterios
 - ▶ Evitar sobrecargar recursos críticos,
 - ▶ asignar tareas dependiendo de habilidades de los agentes,
 - ▶ conseguir solapamiento en responsabilidades para conseguir coherencia global
 - ▶ asignar tareas interdependientes a agentes con proximidad semántica o espacial para minimizar costes de comunicación y sincronización y
 - ▶ reasignar tareas, si es necesario, para completar tareas urgentes.

Distribución de tareas

Mecanismos

- Protocolo de red de contratos
- Negociación mediante mecanismos de mercado como subastas: las tareas son asignadas a agentes mediante acuerdo generalizado o selección mutua.
- Planificación multi-agente: los agentes encargados de la planificación los que determinan a qué agentes se asignan qué tareas.
- Estructura organizativa: los agentes tienen responsabilidades fijas y, por lo tanto, tienen tareas fijas asignadas.

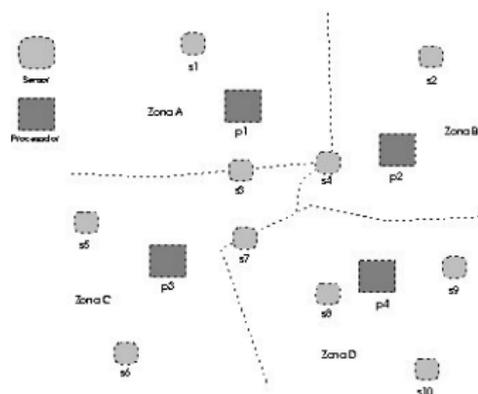
Protocolo de red de contratos

- PRC (Smith,1988 [8]).
- Desarrollado para DPSs, sigue usandose extensivamente en los SMAs
- Sirve para que un agente **contrate** tareas a otros agentes
- Suposiciones
 - ① la negociación es un proceso local que no implica control centralizado,
 - ② existe un medio bidireccional intercambiar información,
 - ③ cada parte en la negociación evalúa la información desde su propia perspectiva
 - ④ el acuerdo final se alcanza mediante selección mútua.

- Dos roles diferentes, manager y contractor
- Manager
 - ▶ inicia la negociación
 - ▶ monitoriza ejecución
 - ▶ procesa los resultados
- Contractor
 - ▶ Se compromete con la realización
 - ▶ ejecuta la tarea
 - ▶ produce los resultados

Ejemplo: DSS (Smith, 1988 [8])

- Se trata de un sistema de sensores distribuido (DSS)
- Consiste en una red de nodos de sensores distribuidos por un área geográfica amplia. No procesan datos, los adquieren.
- Hay otra de nodos procesadores, que han de recibir datos de los nodos sensores. Procesan datos de los nodos de sensores.
- La tarea es la construcción y mantenimiento de un mapa dinámico del tráfico en el área ocupada por la red.



Ejemplo (II)

- Mediante PRC conseguimos que la red de sensores se configure para realizar la tarea (señalización) de forma adecuada.
- La tarea de señalización recoge los datos de todos los sensores y extrae las señales para configurar en tiempo de ejecución el mapa de tráfico
- Managers: nodos procesadores
- Contractors: nodos de señalización

Anuncio de tarea

Cada manager elaborará una propuesta con la tarea de señalización usando un mensaje como el que aparece a continuación:

```
To: * (mensaje broadcast)
From: p2
Type: TASK ANNOUNCEMENT
Contract: 22-3-1
Task Abstraction:
  TASK TYPE SIGNAL
  POSITION LAT 47N LONG 17E
Eligibility Specification:
  MUST-HAVE: SENSOR
  MUST-HAVE: POSITION AREA B
Bid Specification:
  POSITION LAT LONG
  EVERY SENSOR NAME TYPE
Expiration Time:
  28 1730Z FEB 1979
```

Ofertas

El contractor s4 decide que el es adecuado para el anuncio y emite una oferta

To: p2

From: s4

Type: BID

Contract: 22-3-1

Node Abstraction:

POSITION LAT 62N LONG 9W

SENSOR NAME A TYPE S

SENSOR NAME B TYPE S

SENSOR NAME H TYPE T

Concesión del contrato

Una vez que el manager ha decidido entre todas las ofertas, realiza la concesión de la realización de las tareas a aquellos nodos de señalización apropiados, enviando un mensaje como el siguiente:

To: s4

From: p2

Type: AWARD

Contract: 22-3-1

Task Specification:

SENSOR NAME A

SENSOR NAME H

Negociación en SMAs

- Favorece la coordinación y la cooperación entre agentes
- Necesaria tanto cuando los agentes son egoístas como cooperativos. Podemos encontrar una definición
- de negociación entre agentes en que es la siguiente

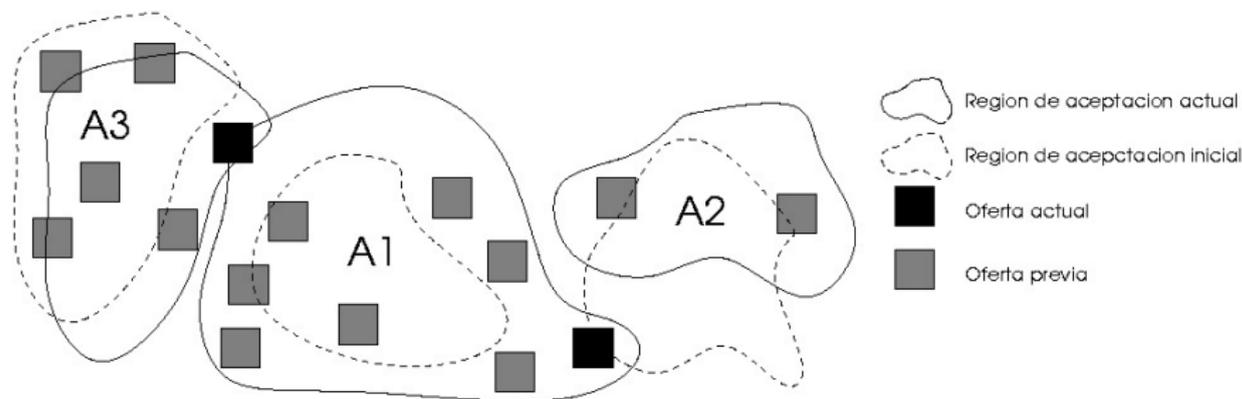
negociación entre agentes es el proceso mediante el cual un grupo de agentes llegan a un acuerdo mutuamente aceptable, sobre algún asunto. (Jennings, 2001 [6])

- Los agentes son autónomos → si quieren en los demás puede que sea necesario negociar
 - ▶ realizar propuestas,
 - ▶ intercambiar opciones,
 - ▶ ofrecer concesiones y, si es posible,
 - ▶ llegar a un acuerdo mutuo aceptable.

Elementos para la negociación

- Protocolos de negociación
 - ▶ roles de agentes,
 - ▶ estados de la negociación,
 - ▶ eventos de transiciones y
 - ▶ mensajes intercambiados
- Objetos sobre los que negociar (i.e. precio de artículo, QoS, términos contractuales, etc)
- modelos usados para la toma de decisiones

Negociación como un proceso de búsqueda



- el agente A_1 negocia con A_2 y A_3
- A_1 puede llegar a acuerdo con A_3 al intersecar sus regiones de aceptación actuales.
- A_1 no puede llegar a un acuerdo con A_2 al no existir intersección

Más detalles de la búsqueda

- Cada uno de los puntos en regiones de aceptación debe tener asignado una puntuación
- Las regiones de aceptación pueden estar sujetas a cambios (desplazamiento, contracción o expansión)
- Proceso negociador más sencillo: Dutch auction
 - ▶ el subastador comunica el precio de subasta del artículo
 - ▶ si no recibe ninguna respuesta, intentará reducir el precio
 - ▶ termina con una respuesta
 - ▶ la única información que llega al subastador para la elaboración de una nueva propuesta es la ausencia de información!!!

Realimentación en la negociación

- El proceso negociador se agiliza si se incluye realimentación (crítica o contraoferta)
- Crítica, ejemplo A

A: Te propongo que me proveas del servicio X bajo ciertas condiciones

B: El precio de X me parece bueno pero el tiempo de suministro es muy pequeño

y ejemplo B

A: Te propongo proporcionarte el servicio Y si tu me proporcionas el servicio X

B: No me interesa el servicio Y

- Contraoferta, ejemplo A

A: Te propongo de que me proveas del servicio X

B: Te propongo proveerte del servicio X si tu me provees del servicio Z

y ejemplo B

A: Te propongo proporcionarte el servicio Y si tu me proporcionas el servicio X

B: Te propongo proporcionarte del servicio X si tu me proporcionas el servicio Z

Conclusiones

- Los sistemas multi-agente suponen una evolución histórica de la IAD
- Los MAS son sistemas en los que el control y los datos están distribuidos entre los agentes
- La comunicación entre agentes se basa en la influencia
- Los MAS usan mecanismos de coordinación complejos



Jacques Ferber.

Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence.

Addison-Wesley, 1999.



Tim Finin, Jay Weber, Gio Wiederhold, Michael Genesereth, Richard Fritson, Donald Mckay, James McGuire, Stuart Shapiro, and Chris Beck.

Specification of the KQML Agent-Communication Language.

NIC, The Darpa Knowledge Sharing Initiative, 1982.



L. C. Lee H. S. Nwana and N. R. Jennings.

Coordination in software agent systems.

The British Telecom Technical Journal, 4(14):79–88, 1996.



N. R. Jennings.

Commitments and conventions: The foundation of coordination in multi-agent systems.

The Knowledge Engineering Review, 8(3):223–250, 1993.



N. R. Jennings.

Coordination techniques for distributed artificial intelligence.

In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. Wiley, 1996.



N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge.

Automated negotiation: prospects, methods and challenges.

Int. J. of Group Decision and Negotiation, 2(10):199–215, 2001.



Yannis Labrou and Tim Finin.

A Proposal for a new KQML Specification.

University of Maryland Baltimore County, 1997.



Reid R. Smith.

The contract net protocol: High-level communication and control in a distributed problem solver.

In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 357–366. Morgan Kaufmann Publishers, Los Altos, CA, 1988.



Michael Wooldridge.

Reasoning about rational agents.

MIT Press, 2000.